P. 1817/88

7
1988

informatyka

Prof. Appelrath o systemie wyszukiwawczym dla płyty wizyjnej

Occam

Techniki obrazowania informacji

WYDAWNICTWO NOT SIGMA

Cena 250 zł

informatyka

7 1988

Nr 7

Miesięcznik Rok XXIII

Lipiec 1988

Organ Komitetu
Naukowo-Technicznego NOT
ds. Informatyki

KOLEGIUM REDAKCYJNE:

Mgr Jarosław DEMINET,
dr inż. Wacław ISZKOWSKI,
mgr Teresa JABŁOŃSKA
(sekretarz redakcji),
Władysław KLEPACZ
(redaktor naczelny),
dr inż. Marek MACHURA,
dr inż. Wiktor RZECZKOWSKI,
mgr inż. Jan RYŻKO,
mgr Hanna WŁODARSKA,
dr inż. Janusz ZALEWSKI,
(zastępca redaktora naczelnego).

PRZEWODNICZĄCY
RADY PROGRAMOWEJ:

Prof. dr hab.
Juliusz Lech KULIKOWSKI

Materiałów nie zamówionych redakcja
nie zwraca

Redakcja: 01-517 Warszawa, ul. Mickiewicza 18
m. 17, tel. 39-14-34

RSW „PRASA-KSIĄŻKA-RUCH"
PRASOWE ZAKŁADY GRAFICZNE
ul. Dworcowa 13, 85-950 BYDGOSZCZ
Zam. 2413/88.
Obj. 4,0 ark. druk. Nakład 9000. E-4

ISSN 0542-9951. INDEKS 36124

Cena egzemplarza 250 zł

WYDAWNICTWO
NACZELNA ORGANIZACJA TECHNICZNA
CZASOPISM I KSIĄŻEK TECHNICZNYCH
SIGMA
00-950 Warszawa
skrytka pocztowa 1004
ul. Biała 4

W NUMERZE: Strona

System wyszukiwawczy dla płyty wizyjnej
Hans-Jurgen Appelrath 1

Język Occam (1)
Sławomir Błaszczak 4

Projektowanie i implementacja wielodostępnego systemu operacyjnego na przykładzie systemu MASTER
Piotr Fuglewicz, Tadeusz Korniak 7

Cyfrowe generowanie obrazów (2)
Jan Zabrodzki 10

Techniki obrazowania informacji o trójwymiarowych zdyskretyzowanych scenach
Wojciech Mokrzycki 13

Nowy sposób pamiętania w systemach informacyjnych
Witold Abramowicz 17

Turbo C. Włączanie warunkowe
Jan Bielecki 19

Lokalna sieć komputerowa D-Link (2)
Andrzej Turowicz 21

ZE ŚWIATA 23

Skrośny system firmy Alsys wspomagania programowania w Adzie dla PC/AT
Nowe bloki systemu Multibus II
Informatyka w Chinach
Kto jest kim w IFIP. Ping-Xin Guo

NOWE KSIĄŻKI 29

TERMINOLOGIA 30

System plików (2)

W NAJBLIŻSZYCH NUMERACH:

- **Władysław M. Turski charakteryzuje metodę kroków kanonicznych, usprawniającą proces opracowania i pielęgnacji programów użytkowych**
- **Frank Foersterling, z Akademii Nauk NRD, omawia wymagania stawiane bazom danych w biurach i charakteryzuje architekturę biurowych systemów informacyjnych**
- **Grzegorz Kaleta i Jacek Owczarczyk porównują efektywność dwóch podstawowych struktur wieloprocesorowych, stosowanych do budowy wieloprocesorowego systemu przetwarzania informacji obrazów**
- **Jan Bielecki podaje przykłady stosowania dodatkowych modyfikatorów języka Turbo C, umożliwiających deklarowanie zmiennych ustalonych i zmiennych nietrwałych**
- **Mariusz Kuc opisuje wywołanie funkcji systemu operacyjnego w programach pisanych w Turbo Pascalu i działających pod kontrolą systemu operacyjnego PC-DOS lub MS-DOS**

HANS-JÜRGEN APPELRATH
Institut für Informatik
Eidgenossische Technische Hochschule
Zurych

POLITECHNIKA ŚLĄSKA Biblioteka Główna w Gliwicach
P. 1817/88

System wyszukiwawczy dla płyty wizyjnej

W dyskusjach na temat technologii pamięci komputerowych coraz większego znaczenia nabierają rozważania dotyczące pamięci zewnętrznych. Mimo dalszego postępu w technice zapisu nie należy spodziewać się znaczących jakościowo zmian w rozwoju pamięci magnetycznych. Można oczekiwać natomiast postępu w technologii i zastosowaniach pamięci (elektro-) optycznych.

Nowoczesne pamięci optyczne znajdują nie tylko coraz więcej zastosowań w elektronicznym sprzęcie powszechnego użytku, ale zyskują coraz większe znaczenie w systemach informacyjnych. Warto tu wymienić dyski kompaktowe (ang. compact disc – CD), cyfrowe płyty optyczne (ang. digital optical recording – DOR) oraz laserowe płyty wizyjne.

W technice CD na zaskakująco małej powierzchni mieści się ogromna pamięć o pojemności 600 MB. Ponieważ jednak pamięci CD umożliwiają tylko odczytywanie zapamiętanej informacji, są więc stosowane tam, gdzie nie jest wymagana aktualizacja zgromadzonej informacji (np. w archiwach, bibliotekach, muzeach). Zapisywanie informacji w pamięci CD jest możliwe tylko metodami laboratoryjnymi.

Pamięć DOR jako pamięć cyfrowo-optyczna umożliwia zapamiętywanie do 1 mln stron formatu A4, co odpowiada pamięci 2 GB. Zapisywanie, pamiętanie i odczytywanie informacji jest kontrolowane przez jedno urządzenie. Proces ten jest nazywany DRAW (ang. direct read after write).

Zainteresowanych bardziej szczegółowo tą problematyką można odesłać [5].

LASEROWA PŁYTA WIZYJNA

Laserowa płyta wizyjna (ang. optical disc) jest pamięcią analogową przeznaczoną do archiwowania pojedynczych kolorowych obrazów oraz sekwencji, a więc także ruchomych obrazów [4, 6]).

Dzięki pracom badawczym i rozwojowym, trwającym już ponad 10 lat, stało się możliwe zastosowanie laserowych płyt wizyjnych w urządzeniach powszechnego użytku oraz w profesjonalnych systemach informacyjnych. Na rynku światowym znane są obecnie trzy systemy laserowych płyt wizyjnych. Ze względu na różne technologie wytwarzania i związanie płyt z różnymi systemami telewizji kolorowej, nie są one ze sobą zgodne. W dalszej części skupiono się wyłącznie na laserowych płytach wizyjnych zrealizowanych w systemie Laser Vision firmy Philips, pracującym według systemu telewizyjnego PAL.

HANS-JURGEN APPELRATH (ur. w roku 1952) jest profesorem zwyczajnym informatyki Uniwersytetu w Oldenburgu (RFN). Uprzednio kierował grupami badawczymi w Uniwersytecie w Dortmundzie i Politechnice Federalnej w Zurychu. Swoje zainteresowania koncentruje na systemach ekspertowych oraz systemach wyszukiwawczych. Jest autorem kilkudziesięciu prac naukowych. Ostatnio wydał książkę „Von Datenbanken zu Expertensystemen".

Informacja o obrazie jest zapisywana w spiralnych rowkach. Analogowym nośnikiem informacji są wykonane w rowkach wgłębienia (tzw. pits) o zmiennej długości. Wgłębienia są wykonane za pomocą lasera w górnej, światłoczułej warstwie płyty wizyjnej. Każda ze stron laserowej płyty wizyjnej może zawierać do 54 tys. pojedynczych obrazów (jeden obraz w jednym spiralnym rowku płyty), co odpowiada pojemności 2 GB. Warstwa metalowa i umieszczona na niej warstwa przezroczystego sztucznego tworzywa, o grubości 1 mm, tworzą charakterystyczną dla laserowych płyt wizyjnych metalizującą powierzchnię, która chroni płytę przed uszkodzeniami mechanicznymi i termicznymi.

W obecnym stadium rozwoju technologii laserowych płyt wizyjnych nie są możliwe zmiany raz zapisanego obrazu. Ogranicza to znacznie ich zastosowanie. Co prawda różni producenci pracują nad wielokrotnie zapisywalną płytą wizyjną, jednak poglądy na temat terminu, w którym taka płyta pojawi się na rynku, są rozbieżne.

Odczytywanie informacji z płyty wizyjnej odbywa się w odtwarzaczu, który wysyła promienie laserowe w kierunku płyty. W ten sposób identyfikuje się konfigurację wgłębień w rowkach. Informacja zostaje przetworzona na obraz, który może być pokazany na kolorowym monitorze. Płyta w odtwarzaczu obraca się z szybkością 1500 obrotów na minutę. Czas dostępu do dowolnego obrazu wynosi od 2 do 5 sekund. Obrazy mogą być wyszukiwane w dowolnej kolejności. W trakcie prezentowania na monitorze pojedynczego obrazu promień laserowy „odczytuje" pojedynczy rowek. Pokazując sekwencję nieruchomych obrazów odtwarzacz kieruje promień laserowy do coraz to innych rowków, w których są zapamiętywane kolejne obrazy sekwencji. Szybkość obrotu płyty umożliwia pokazywanie na monitorze obrazów z wymaganą przez fizjologię oka ludzkiego częstotliwością 25 obrazów na sekundę.

Obrazy mogą być zgrupowane w tzw. rozdziały (do 78 rozdziałów na jednej stronie płyty). Obrazy i rozdziały są identyfikowane przez numery. Użytkownik może wybierać je także przy użyciu zdalnego sterowania (tzw. pilota): dowolny obraz lub rozdział przez podanie jego numeru, następny lub poprzedni obraz względem już przeczytanego. Użytkownik może także przeglądać dowolną sekwencję z szybkością większą (dla pobieżnego zorientowania się w informacji) lub mniejszą (dla dokładniejszego zapoznania się ze szczegółami obrazu). Dowolne obrazy lub rozdziały można chronić przed niepożądanym dostępem.

KOMPUTEROWO WSPOMAGANE WYKORZYSTANIE INFORMATYKI

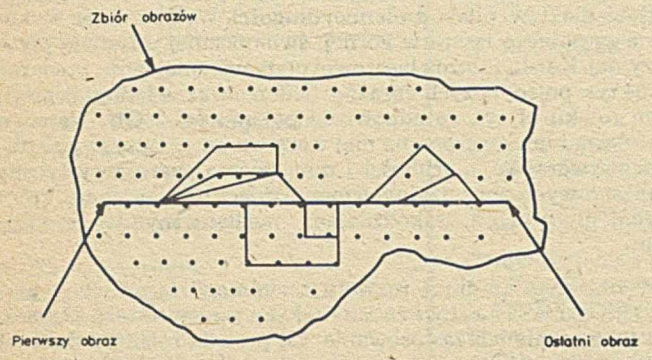
Laserowa płyta wizyjna umożliwia wyszukiwanie pożądanej informacji z użyciem opisanych funkcji. Pomocny może być także oddzielnie przygotowany katalog, zawierający informacje o zgromadzonych na płycie obrazach. Takie użytkowanie laserowej płyty wizyjnej wyklucza jednak bardziej złożone zastosowania wymagające analizy zawartości informacyjnej zgromadzonych obrazów.

Dzięki połączeniu odtwarzacza laserowych płyt wizyjnych z komputerem jest możliwe:

- zrealizowanie działającego szybko i efektywnie systemu wyszukiwawczego,
- tworzenie inteligentnych systemów, dla których informacje

zawarte w zgromadzonych obrazach stanowią całość lub część wiedzy systemu (przede wszystkim do zastosowań edukacyjnych).

Przy kształceniu wspomaganym komputerem z zastosowaniem płyty wizyjnej podstawowym zadaniem jest określenie sekwencji obrazów, które następnie w trakcie dialogu z użytkownikiem będą w sposób wskazany prezentowane na monitorze. Ważne jest właściwe oprogramowanie wariantów scenariuszy umożliwiających prezentowanie różnych sekwencji obrazów. Na rys. 1 zaznaczono scenariusze wiodące od jednego obrazu do drugiego (pojedyncze obrazy przedstawiono jako punkty). Jak widać, użytkownik między obejrzeniem pierwszego i ostatniego obrazu może zobaczyć różne ich sekwencje. Kolejność przeglądania jest jednak poważnie ograniczona. Sterują nią struktury oprogramowania wynikające z przyjętych metod dydaktycznych.



Rys. 1. Przykładowe przejścia od obrazu do obrazu – obrazy są reprezentowane przez punkty

W zastosowaniach wyszukiwawczych trudno jest określić stałe scenariusze przeszukiwania zgromadzonych obrazów. Pożądane przez użytkownika zbiory obrazów zostają określone na podstawie zapytań boolowskich lub w drodze wykonywania teoriomnogościowych operacji na wcześniej określonych zbiorach obrazów. Dlatego niezbędne jest uzupełnienie każdego ze zgromadzonych obrazów o deskryptory opisujące zawartość informacyjną obrazu (indeksowanie). Deskryptory mogą być uzupełnione lub zastąpione informacjami bibliograficznymi obrazu, takimi jak: nazwisko autora, data powstania obrazu. Deskryptory i informacje bibliograficzne są następnie wykorzystywane przez algorytmy wyszukiwacze.

ARCHITEKTURA SYSTEMU WYSZUKIWAWCZEGO

W architekturze systemów wyszukiwawczych z laserową płytą wizyjną można wyróżnić trzy składniki nośnika informacji:

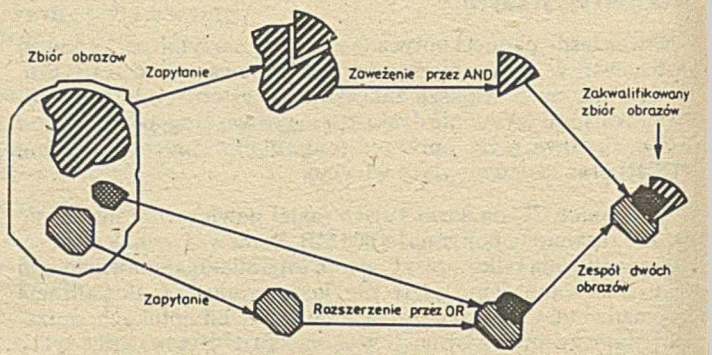
- Moduł podstawowy systemu odpowiada za sterowanie odtwarzaczem laserowych płyt wizyjnych, zapewnia wyszukiwanie obrazu o określonym numerze i przekazuje go na monitor. Podstawowy moduł systemu powinien zapewnić taką samą komunikację z pozostałymi modułami bez względu na typ odtwarzacza.
- Moduł wyszukiwawczy określa zbiór obrazów stanowiących odpowiedź na zapytanie użytkownika. Wyszukiwanie odbywa się na podstawie analizy deskryptorów i informacji bibliograficznych o zgromadzonych obrazach.
- Moduły aplikacyjne stanowią oprogramowanie związane ze specyfiką zastosowania systemu. Umożliwiają sterowanie dialogiem z użytkownikiem i komunikują się z modułem wyszukiwawczym. Wymagają przy tworzeniu ścisłej współpracy informatyków ze specjalistami z dziedziny zastosowań.

System wyszukiwawczy ODIR

System wyszukiwawczy ODIR (ang. optical disc information retrieval) powstał w Instytucie Informatyki Politechniki Federalnej w Zurychu, w grupie CH-MED. Podjęte prace umożliwiły [7]:

– wyprodukowanie pierwszej szwajcarskiej laserowej płyty wizyjnej z dziedziny medycyny,
– ocenę możliwości zastosowań odtwarzacza płyt wizyjnych w połączeniu z komputerem osobistym,
– opracowanie metod tworzenia systemów informacyjnych z zastosowaniem płyty wizyjnej.

Na rys. 1. przedstawiono sposób prezentacji obrazów zgromadzonych na płycie, jako realizację pewnego scenariusza. W systemach wyszukiwawczych działają natomiast algorytmy znajdujące odpowiedź na zapytania boolowskie (deskryptory połączone operatorami logicznymi AND, OR, AND NOT itp.) jako podzbiór zbioru wszystkich obrazów zgromadzonych na płycie wizyjnej. Znaleziony podzbiór może zostać w połączeniu z uprzednio określonymi podzbiorami poddany działaniu operatorów teoriomnogościowych. Podzbiór może być także zawężony przez działanie na nim nowego zapytania boolowskiego. Operacje te mogą być tak długo powtarzane, aż podzbiór obrazów będzie spełniał wymagania jakościowe (znalezione to, co szukane) i ilościowe (np. moc podzbioru nie jest zbyt duża) postawione przez użytkownika. Proces wyszukiwania zilustrowano na rys. 2.



Rys. 2. Ilustracja procesu wyszukiwania obrazów

System ODIR został zimplementowany w języku Pascal na IBM PC/XT. Umożliwia on:

- formułowanie zapytań boolowskich i znajdowanie podzbioru obrazów będącego odpowiedzią na zapytania,
- wykonywanie operacji teoriomnogościowych na zbiorze obrazów (iloczyn, suma i różnica zbiorów),
- manipulowanie podzbiorem przez usuwanie i wprowadzanie pojedynczych obrazów,
- prezentację podzbioru obrazów na monitorze.

ODIR został także zimplementowany w języku Modula-2 na komputerach Lilith [1] oraz MacIntosh. Implementacje te potwierdziły słuszność tych rozwiązań z dziedziny inżynierii oprogramowania, które umożliwiły m. in. przenoszenie oprogramowania. Pozwoliły one na lepsze rozwiązanie dialogu człowieka z komputerem, przez szersze zastosowanie grafiki oraz sterowanie dialogu za pomocą myszy.

Na rys. 3 przedstawiono drzewiastą strukturę menu, co ilustruje działanie systemu. Początkowe menu (menu 0) umożliwia zarządzanie wyszukiwanymi zbiorami obrazów: zapamiętanie ich adresów – S, wyszukiwanie uprzednio zapamiętanych – L, usunięcie uprzednio zapamiętanych – D oraz inicjalizację list menu znajdujących się na niższym poziomie drzewa.

Menu 1.0 steruje formułowaniem zapytań boolowskich. Możliwe są trzy rodzaje zapytań.

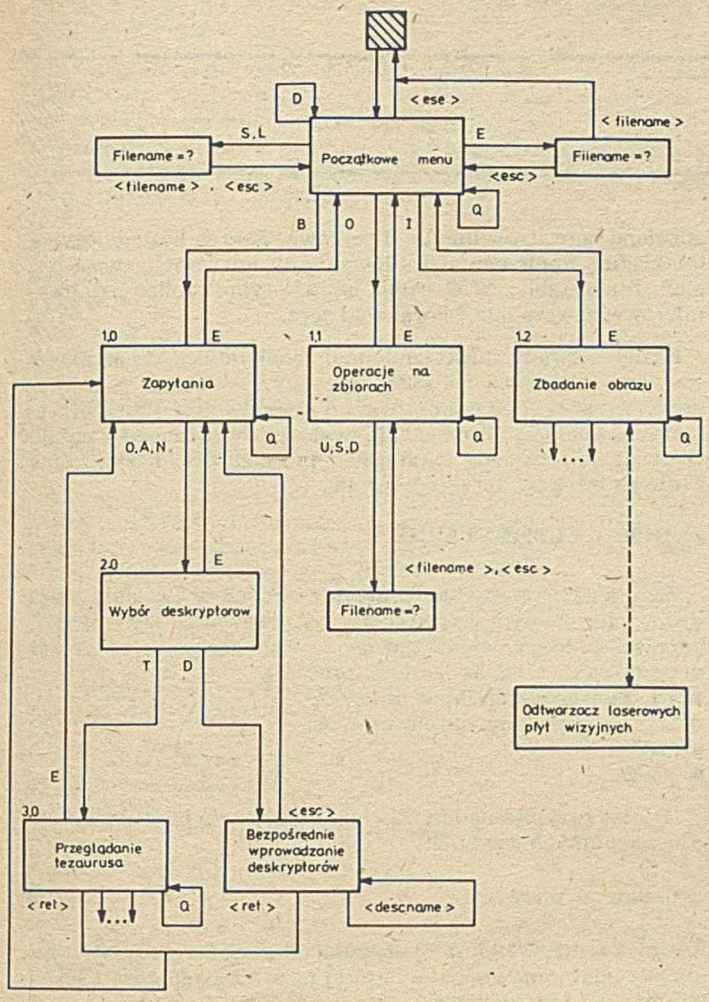
(deskryptor1) OR (deskryptor2) OR ... (deskryptorN)
(deskryptor1) AND (deskryptor2) AND ... (deskryptorN)
(deskryptor1) AND NOT (deskryptor2)

Sposób wybierania deskryptorów jest określony w menu 2.0. Istnieją dwa sposoby wyboru deskryptora:

- wybór w tezaurusie odpowiednich deskryptorów (menu 3.0);
- wprowadzenie nazw deskryptorów z klawiatury.

Po wyszukaniu zbioru obrazów relewantnych dla zapytania system powraca automatycznie do menu 1.0 oraz pokazuje moc wyszukiwanego zbioru (która w systemach wyszukiwawczych

często jest określana jako syntentyczny wynik wyszukiwania). Wyszukany zbiór relewantnych obrazów może być zmieniany przez określenie na nim podzbioru. Jest to możliwe przez sformułowanie zapytania działającego na wyszukanym zbiorze.



Rys. 3. Struktura menu systemu ODIR

Polecenia z menu 1.1 umożliwiają wykonywanie operacji iloczynu, sumy i różnicy między wyszukiwanym zbiorem obrazów a jakimś innym wskazanym zbiorem. Polecenia z menu 1.2 umożliwiają przeglądanie (ang. browsing) w różny sposób wyszukiwanego zbioru obrazów oraz usuwanie ze zbioru wskazanych obrazów. Dotychczasowe doświadczenia z systemem wskazują, że umożliwia on nawet niewprawnemu użytkownikowi szybkie i efektywne wyszukiwanie. Oczywiście istnieje wiele możliwości dalszego udoskonalenia wyszukiwania.

Koncepcja ulepszonej strategii wyszukiwania

Porównanie komputerowo wspomaganego wyszukiwania obrazów zgromadzonych na laserowych płytach wizyjnych z klasycznymi komputerowymi systemami wyszukiwania tekstów (głównie opisów bibliograficznych) pozwala na sformułowanie następujących stwierdzeń:

- Fizyczne sąsiedztwo obrazów na laserowej płycie wizyjnej implikuje, w przeciwieństwie do klasycznych systemów wyszukiwania tekstów, pokrewieństwo znaczeniowe obrazów. Właściwość ta powinna zostać wykorzystana w określeniu strategii wyszukiwania.
- Przeglądanie obrazów zgromadzonych na płycie wizyjnej jest znacznie szybsze niż tekstów w systemach wyszukiwawczych. Stwierdzenie, czy jakiś obraz jest relewantny czy nie, trwa z reguły krócej niż 1 sekundę (ocena tekstu trwa znacznie dłużej).

Stwierdzenia te upoważniają do rezygnacji z opisywania pojedynczych obrazów własnymi deskryptorami. Obrazy te mogą być grupowane w klasy i tylko klasy powinny być indeksowane. Doświadczenie wskazuje, że jeżeli zostanie wyszukanych 200 relewantnych obrazów, to mogą być one w bardzo krótkim czasie ocenione przez użytkownika. W ten sposób można efektywnie zawęzić zbiór relewantnych obrazów.

W systemach wyszukiwawczych tekstu natomiast nie analizuje się wyszukiwanych dokumentów, kiedy ich zbiór wydaje się zbyt liczny. Tworzy się nowe zapytania zawężające zbiór relewantnych dokumentów.

Zaletami tej nowej koncepcji wyszukiwania są:
- oszczędność czasu w indeksowaniu obrazów deskryptorami, a także minimalizacja błędów indeksowania (są one nie do uniknięcia),
- oszczędność pamięci,
- przyspieszenie procesu wyszukiwania.

Ostatnie dwa czynniki umożliwiają implementację efektywnych systemów wyszukiwawczych z wykorzystaniem komputerów klasy IBM PC/XT

* * *

Laserowe płyty wizyjne, jako elektroniczno-optyczna pamięć, tworzą nowe perspektywy zastosowań w zakresie systemów wyszukiwawczych. Są one szczególnie interesujące, jeżeli odtwarzacz płyt jest sterowany przez komputer osobisty. Są to urządzenia wzajemnie uzupełniające się, jeżeli uwzględni się, że na płytach wizyjnych zgromadzone są obrazy, a w komputerze dane opisujące je i algorytmy efektywnego ich wyszukiwania.

Tłum. i oprac.:
WITOLD ABRAMOWICZ

LITERATURA

[1] Abramowicz W.: Modula-2 i Lilith – zgodność metod i narzędzi informatycznych. Informatyka, nr 4, str. 1–5, 1984
[2] Appelrath H. J.: ODIR – Optical disc information retrieval. Datenbank-Systeme für Büro Technik und Wissenschaft, GI-Fachtagung, Karlsruhe, März 1985
[3] BYTE, No. 7, pp. 149–228, July 1984
[4] Computer Magazin, Nr. 4, s. 24–32, April 1986
[5] Fujitani L.: Laser optical disc – the coming revolution in online storage. Communications of the ACM, Vol. 27, No. 6, pp. 546–554, June 1984
[6] Furrer G.: Interaktive audiovisuelle Informationssysteme mit Bildplatte und Computer. Technische Rundschau, Nr. 6, 1984
[7] Interessengemeinschaft CH-MED Laserbildplattenarchiv. Projektbeschriebung. Zurich, Februar 1984.

Z przykrością informujemy Czytelników, że począwszy od numeru lipcowego br. cena INFORMATYKI została ustalona na 250 zł za egzemplarz.

Nowa cena obowiązuje nowych prenumeratorów oraz sprzedaż odręczną. Jednak nie wyklucza się ewentualnej dopłaty różnicy (pomiędzy nową a starą ceną) do już opłaconej prenumeraty na drugie półrocze br. O ewentualnej dopłacie prenumeratorzy zostaną powiadomieni pisemnie przez Dział Kolportażu Wydawnictwa SIGMA. Dotychczasowa cena ulgowa – 50 zł – zostaje utrzymana w prenumeracie do końca 1988 roku.

Jednocześnie informujemy, że od stycznia 1989 roku cena INFORMATYKI będzie wynosić:
300 zł za egzemplarz – cena normalna
60 zł za egzemplarz – cena ulgowa.

Szczegóły podano w „Warunkach prenumeraty" na s. 26.

SŁAWOMIR BŁASZCZAK
Instytut Informatyki
Politechnika Warszawska

Język Occam (1)

Systemy wieloprocesorowe odgrywają i – jak należy przypuszczać – będą odgrywały dominującą rolę w rozwoju informatyki. Istotnym problemem związanym z takimi systemami jest to, jak ma wyglądać ich oprogramowanie, aby można było w pełni i efektywnie wykorzystać ich zasoby. Pewną próbą rozwiązania tego problemu była koncepcja przedstawiona przez Hoare'a [2]. Przyjął on model systemu, w którym występuje zbiór procesów sekwencyjnych przesyłających między sobą komunikaty. Hoare zaproponował tam również pewną notację do opisu takich systemów (język CSP).

Rozwinięciem tej koncepcji jest język Occam [3, 4], stworzony w ramach przedsięwzięcia realizowanego przez firmę INMOS. Projekt Occamu powstał równolegle z projektem procesorów zwanych transputerami, które miały być oprogramowywane w tym języku. Transputery są jednoukładowymi komputerami wyposażonymi dodatkowo w cztery sprzęgi, przez które mogą się komunikować z czterema takimi samymi układami (por. [1]). Proces przesyłania informacji jest niezależny od akcji jednostki centralnej. Program w Occamie, który – tak jak w języku CSP – składa się ze współbieżnych i współpracujących ze sobą procesów, dokładnie odpowiada takiej strukturze połączonych transputerów. Z drugiej strony Occam, w swoich założeniach, nie jest przeznaczony dla żadnego konkretnego komputera. Można go użyć w dowolnym systemie, również w jednoprocesorowym, o ile istnieje odpowiedni kompilator i środowisko programowe.

Occam jest językiem wysokiego poziomu, a programy są w nim zapisywane w sposób strukturalny. Zbiór dostępnych typów danych oraz metod ich strukturalizacji jest jednak dość ubogi.

W artykule przedstawiono język Occam w sposób pełny, choć nie wprowadzono formalnych definicji. Podane przykłady powinny wyjaśnić wątpliwości tam, gdzie opis słowny pomija niektóre szczegóły składni.

PROCESY

Podstawowym pojęciem języka Occam jest proces. Jego definicja jest rekurencyjna: proces jest złożeniem procesów podległych, te z kolei składają się z innych procesów, bądź są procesami elementarnymi. Sposób i kolejność realizowania procesów podległych wynika z zastosowanej konstrukcji i może być: sekwencyjny, równoległy itp. Procesy elementarne (ang. primitive processes) są z reguły realizowane przez oprogramowanie jądra przy użyciu istniejącego sprzętu.

Ogólny model przetwarzania danych stosowanych w Occamie można przedstawić następująco:
- odbierz dane z innej części systemu (innego procesu),
- przekształć dane (wykonaj obliczenia),
- wyślij wynik dalej.

Informacje są przekazywane kanałami. Każdy kanał może łączyć tylko dwa procesy, a informacje mogą być nim przesyłane tylko w jednym kierunku. Dane do obliczeń i wyniki przechowuje się w zmiennych. W obecnej wersji Occamu istnieją tylko dwa proste typy danych – słowo i bajt, oraz jeden typ strukturalny – tablica. Słowo (**bajt**) może zawierać liczby całkowite przedstawione jako uzupełnienie dwójkowe. Zakres liczb zależy od konkretnej implementacji. Słowa, bajty lub kanały mogą być elementami tablic. W Occamie istnieją tylko tablice jednowymiarowe, zawsze indeksowane od zera.

Każdy używany obiekt (zmienna, kanał, tablica i inne, przedstawione dalej) musi być zadeklarowany. Deklaracje są związane z jednym konkretnym procesem i tylko w nim (lub w procesach mu podległych) można używać wprowadzonego obiektu. Obowiązują tu zasady lokalności i przesłaniania nazw, znane z innych języków oprogramowania.

PROCESY ELEMENTARNE

Poszczególne etapy przetwarzania danych w Occamie mogą być w najprostszym wypadku zrealizowane przy użyciu następujących procesów elementarnych:
- przypisania (ang. assignment process),
- wejściowego (ang. input process),
- wyjściowego (ang. output process),
- WAIT,
- SKIP.

Proces przypisania ma składnię podobną do instrukcji przypisania z innych języków:

zmienna := wyrażenie

Również semantyka jest identyczna; obliczona wartość wyrażenia jest nadawana zmiennej, której nazwa znajduje się z lewej strony znaku przypisania.

Pewne niewielkie różnice pomiędzy Occamem a innymi językami występują w definicji wyrażenia. Wyrażenie jest to ciąg elementów połączonych operatorami. Elementem może być zmienna (w tym również zmienna indeksowana) lub stała liczbowa. Stałe mogą być zapisane w postaci dziesiętnej, szesnastkowej (przy czym cyfry są poprzedzone znakiem # , np. # F6) lub znakowej (znak jest w apostrofach). Można zapisać zarówno znaki widoczne (np. 'a'), jak i sterujące (np. '*n' lub '* # OA'). W tym ostatnim wypadku gwiazdka poprzedza kod znaku lub specjalny symbol znaku sterującego (np. n – nowa linia).

Operatory dzielą się na:
- arytmetyczne – dodawania (+), odejmowania (−), mnożenia (×), dzielenia (/), brania modulo (|);
- logiczne – iloczynu (AND), sumy (OR), negacji (NOT);
- działające na bity – iloczyn (∧), suma (∨), suma modulo 2 (><), przesunięcie w prawo (<<) i w lewo (>>);
- porównań (=, <>, <, >, <=, >=).

Operacje arytmetyczne dają zawsze wynik całkowity. Operatory logiczne traktują argumenty jako wartości logiczne, przy czym liczba różna od zera jest traktowana jako wartość „prawda". W wyrażeniach można używać stałych TRUE (wszystkie bity równe 1) i FALSE (wszystkie bity równe Ø). Operacje logiczne iloczynu i sumy różnią się od operacji na bitach tym, że te pierwsze zawsze udostępniają wartości TRUE lub FALSE (same 1 lub 0), podczas gdy w drugim wypadku jest to iloczyn lub suma poszczególnych bitów argumentów. Ponadto w operacjach logicznych, jeśli wartość prawego argumentu przesądza o wyniku całej operacji (FALSE dla AND lub TRUE dla OR), to lewy argument nie jest obliczany. Na przykład, w wyrażeniu:

(i>0) AND (i<MAX) AND (Tabl[i] = 'a')

jeśli indeks i będzie poza zakresem tablicy, to trzecie podwyrażenie nie będzie obliczone, co zapobiega błędowi wykonania.

W Occamie operatory nie mają priorytetów; wyrażenia są obliczane od strony lewej do prawej. Kolejność tę można zmienić stosując nawiasy okrągłe.

Do komunikacji między procesami służą procesy wejściowe i wyjściowe. W ich zapisie podaje się nazwę kanału, przez który odbywa się komunikacja, oraz wyrażenie, którego wartość jest wysyłana lub nazwę zmiennej, za którą zostanie podstawiona odebrana liczba. Te dwa elementy są rozdzielone znakiem ! (dla wyprowadzania) lub ? (dla wprowadzania).

Przykłady:

```
kan1 ! x + 5
kan1 ? z
dane ? zmienna
wynik ! zmienna + 2
```

Za znakiem rozdzielającym (!, ?) można również umieścić słowo kluczowe ANY; nie jest wówczas istotna przesyłana treść, a jedynie sam fakt komunikacji.

Dla danego kanału tylko jeden proces spośród wszystkich działających równolegle może realizować wyprowadzanie i tylko jeden inny może oczekiwać na przyjęcie komunikatu. Operacja komunikacji procesów jest w pełni synchroniczna, tzn. każdy z pary czeka, aż drugi wykona operację komplementarną. Następnie, po wymianie informacji, procesy wykonują swoje dalsze akcje równolegle.

Przy przesłaniach dłuższych ciągów informacji niż jednosłowowe można użyć zapisu skróconego:

```
kanał ! wyr1; wyr2; wyr3; ...
kanał ? zm1; zm2; zm3; ...
```

co jest równoważne sekwencji procesów:

```
kanał ! wyr1
kanał ! wyr2
...
kanał ? zm1
kanał ? zm2
...
```

Proces WAIT służy do wstrzymywania działania procesu, w skład którego wchodzi, do chwili spełnienia podanego warunku czasowego. Na przykład, przy sekwencyjnej realizacji następujących procesów:

```
time := NOW + 60
WAIT NOW AFTER time
sygnał ! ANY
```

Wyprowadzanie sygnału przez kanał będzie wykonane po 60 jednostkach czasu od chwili wykonania pierwszego procesu. Zmienna NOW, dostępna tylko do odczytania, podaje lokalny czas procesu. W Occamie nie istnieje pojęcie czasu globalnego dla całego systemu; każdy równolegle działający komponent może mieć swój własny, niezależny zegar. Warunek czasowy procesu WAIT określa się wyrażeniem:

NOW AFTER wyrażenie

Warunek czasowy jest spełniony, gdy zmienna NOW osiągnie wartość równą lub większą niż wartość podanego wyrażenia.

Ostatni z procesów elementarnych, proces SKIP, jest równoważny akcji pustej i odpowiada instrukcji bezczynności z innych języków.

PROCESY ZŁOŻONE

Wymienione procesy elementarne mogą być użyte do tworzenia bardziej złożonych konstrukcji (ang. constructs). Każda taka konstrukcja stanowi nowy proces i może stać się elementem innej konstrukcji. Konstrukcja rozpoczyna się odpowiednim słowem kluczowym (tzw. konstruktorem), a w jej zapisie mogą się znaleźć dodatkowe frazy pomocnicze. Stosuje się przy tym specjalne zasady formatowania tekstu. Przyjmuje się, że jeden wiersz może zawierać co najwyżej zapis jednego procesu elementarnego, konstruktora lub frazy. Ponadto zapis procesów wchodzących w skład konstrukcji musi rozpoczynać się przynajmniej dwie kolumny dalej w prawo (z wcięciem) niż słowo kluczowe lub fraza tworząca konstrukcję. Na przykład:

```
SEQ          -- słowo kluczowe
  in ? x     -- procesy podległe
  out ! x
```

W ten sposób same reguły składni języka wymuszają strukturalne podejście do tworzenia i zapisu programów. Zwiększa to niewątpliwie ich czytelność, ale jednocześnie utrudnia wprowadzanie.

W powyższym przykładzie pokazano również sposób komentowania programu; tekst umieszczony między dwoma myślnikami a końcem wiersza nie jest analizowany i może być użyty do objaśnień.

Konstrukcja nie tylko grupuje pewien zbiór procesów podległych, lecz także określa sposób ich realizacji (sekwencyjny, równoległy itp.). We wszystkich wypadkach proces złożony kończy się, gdy zakończą się wszystkie wybrane do wykonania procesy podległe.

Z każdym procesem złożonym może być związana deklaracja nazw w nim używanych. Deklaracje są zapisywane w wierszach poprzedzających konstruktora inicjującego proces, począwszy od tej samej kolumny, co ten konstruktor.

Deklaracja zmiennych i tablic rozpoczyna się słowem kluczowym VAR, po którym umieszcza się listę nazw. Za nazwą tablicy musi być podany jej rozmiar wyrażony w słowach lub bajtach, np.:

```
VAR zm, x, i, tab [10], tab.bajtowa [BYTE 30], ...
```

Kropka jest dopuszczalnym znakiem nazwy.

Podobnie wygląda deklaracja kanałów. Tym razem lista nazw jest poprzedzona słowem CHAN, np.:

```
CHAN kan1, in, out, tab.kan [10], ...
```

W Occamie można również nadawać nazwy stałym. W tym wypadku, za słowem kluczowym DEF podaje się listę definicji stałych, każda w postaci:

nazwa = wyrażenie

Wyrażenie musi być obliczalne podczas kompilacji, np.:

```
DEF n = 100, m = n + 1, litera = 'a', ...
```

Z jedną nazwą można wiązać również całą tablicę stałych, np.:

```
DEF kod = TABLE [10, 20, 21, 32]
    tab.bajt = TABLE [BYTE # 1A, # 1F, 0], ...
```

Słowo BYTE, podane za nawiasem kwadratowym otwierającym w deklaracji tablicy zmiennych lub stałych, oznacza, że jej elementy zajmują jeden bajt. Do elementów tablic stałych można się odwoływać podając nazwę tablicy i indeks (jak do zmiennych indeksowych). Jeśli tablica stałych ma zawierać tekst, to można zastosować zapis skrócony, np.:

```
tekst = "to jest tekst"
```

co jest równoważne definicji tablicy bajtowej, której element pierwszy (o indeksie 0) zawiera długość tekstu (maksymalnie 255 znaków), a w pozostałych są umieszczone kody kolejnych znaków.

Każda deklaracja lub definicja poprzedzająca proces musi być zakończona znakiem dwukropka, np.:

```
CHAN ch1, ch2:
PAR
  VAR x:
  SEQ
  ...
```

Procesy, których wykonanie ma być sekwencyjne, tzn. następny rozpoczyna się po zakończeniu procesu poprzedniego, są grupowane za pomocą konstruktora SEQ, np.:

```
VAR x:
SEQ
  c1 ? x
  x := x + 1
  c2 ! x
```

Konstruktor PAR z kolei pozwala określić zbiór procesów, które mogą się wykonywać współbieżnie. Na przykład proces:

```
CHAN ch, ch1 :
PAR
  VAR x, y:
  SEQ       -- proces pierwszy
    in ? x; y
    ch ! x
    y := y + 1
    ch1 ? x
    out ! x + y
  VAR x, y:
  SEQ       -- proces drugi
    ch ? x
    ch1 ! x + 1
```

składa się z dwóch procesów sekwencyjnych wykonujących się współbieżnie i współpracujących ze sobą.

Procesy współbieżne mogą się ze sobą komunikować jedynie przez kanały, a wspólne zmienne mogą być tylko odczytywane. Kompilator powinien zapewnić spełnienie tego warunku.

Occam nie określa, w jaki sposób ma być realizowane współbieżne wykonanie procesów. Dwa procesy mogą działać na dwóch różnych procesorach lub dzielić czas jednego. W obu wypadkach wersja źródłowa programu będzie identyczna, zmianie musi ulec tylko system translacji oraz oczywiście musi być dostępny odpowiedni sprzęt.

W języku programowania współbieżnego ukierunkowanym na przesyłanie komunikatów, musi istnieć konstrukcja umożliwiająca równoczesne oczekiwanie na informacje z wielu źródeł. W Occamie służy do tego konstrukcja ALT. Proces:

```
VAR x:
ALT
  in1 ? x
    out ! x
  in2 ? x
    out ! x
```

odbiera informację z kanału **in1** lub **in2** i wysyła ją jednym kanałem **out**, łącząc tym samym logicznie dwa kanały wejściowe.

Proces ALT składa się z sekwencji procesów dozorowanych. Każdy proces dozorowany składa się z dozoru (ang. guard) oraz procesu, np.:

```
in1 ? x
  out ! x
```

Zgodnie z zasadami składni Occamu, dozór i proces muszą być zapisane z użyciem wcięć. Dozór może wystąpić w jednej z trzech następujących postaci:

wyrażenie & wejście
wyrażenie & „wait"
wyrażenie & SKIP

We wszystkich wypadkach wyrażenie wraz ze znakiem & może być pominięte. W dozorach części oznaczone jako **wejście** i „wait" zapisuje się identycznie jako odpowiednie procesy – wejściowy i WAIT. Dozór jest uznawany za gotowy, jeśli wyrażenie (o ile występuje) ma wartość TRUE oraz:

- dla frazy **wejście**, gdy inny proces wysłał komunikat przez wymieniony kanał,
- dla frazy **„wait"**, gdy warunek czasowy jest prawdziwy (tj. minął określony czas),
- dla frazy SKIP – zawsze.

Działanie procesu ALT jest wstrzymywane do chwili, gdy choć jeden z dozorów będzie gotowy. Następnie jest realizowany proces związany z dozorem i cała konstrukcja ALT kończy się. Gdy więcej niż jeden dozór jest prawdziwy, to jest wybierany według sprawiedliwej zasady jeden z nich. W procesie:

```
VAR buf [100] , l, x, lwe, lwy :
SEQ
 l := 0
 lwe := 0
 lwy := 0
 ALT
  l < 100 & prod ? x
    SEQ       -- wstawienie do bufora
      buf [lwe] := x
      lwe := lwe + 1 \100
      l := l + 1
  l > 0 & kon1 ? ANY
    SEQ       -- pobranie z bufora
      kon2 ! buf [lwy]
      lwy := lwy + 1 \100
      l := l - 1
```

gdy $0 < l < 100$, czyli bufor będzie w części zapełniony, to można do niego zarówno wstawiać, jak i pobierać.

Tak jak większość innych języków programowania, również Occam zawiera konstrukcję umożliwiającą wybór jednego z wielu procesów zależnie od podanych warunków (stanu programu) – jest nią proces warunkowy **IF**. Jego składnia jest zbliżona do składni procesu ALT; po słowie kluczowym **IF** może wystąpić ciąg warunków, z każdym jest skojarzony jeden proces. Warunek jest z reguły wyrażeniem logicznym i jeśli jest spełniony, to jest wykonywany podległy mu proces. Na przykład w procesie:

```
IF
  i = 1
    out1 ! x
  i = 2
    out2 ! x
```

wartość zmiennej **x** będzie wysłana przez kanał **out1**, jeśli **i = 1**, lub przez **out2**, gdy **i = 2**. Jeśli żaden z warunków procesu **IF** nie jest spełniony, to jest on kończony natychmiast; jeśli natomiast więcej niż jeden jest prawdziwy, to jest wybierany tekstowo pierwszy, a pozostałe są ignorowane.

Ostatnia z konstrukcji Occamu, proces WHILE, umożliwia tworzenie pętli. Proces podległy procesowi WHILE będzie wykonywany do chwili, gdy wyrażenie logiczne zapisane za słowem WHILE będzie równe FALSE, na przykład:

```
VAR x :
SEQ
  x := 0
  WHILE x >= 0
    SEQ
      c1 ? x
      c2 ! x
```

Powyższy proces kopiuje wartości z kanału **c1** do **c2** do chwili, gdy **x<0**.

Konstrukcja WHILE jest bardzo często używana do zapisywania procesów działających cyklicznie w pętli nieskończonej. W tym wypadku wyrażenie redukuje się do stałej TRUE. Na przykład poniższy proces sumuje dwie, odebrane liczby i wynik przekazuje dalej:

```
WHILE TRUE
  VAR x, y:
  SEQ
    in ? x; y
    out ! x + y
```

LITERATURA

[1] Caban D.: Transputer – narzędzie przetwarzania równoległego. Informatyka, nr 6, 1988

[2] Hoare C. A. R.: Communicating sequential processes. Comm. ACM, Vol. 21, No. 8, pp. 666–667, 1978

[3] INMOS: Occam Programming Manual. Series in Computer Science, Prentice-Hall, 1984

[4] May D.: Occam. SIGPLAN Notices, Vol. 18, No. 4, pp. 64-65, 1983.

PIOTR FUGLEWICZ
TADEUSZ KORNIAK
Centrum Naukowo-Produkcyjne
Systemów Sterowania MERASTER
Katowice

Projektowanie i implementacja wielodostępnego systemu operacyjnego na przykładzie systemu MASTER

Realizacja systemu operacyjnego od podstaw jest zjawiskiem niezbyt częstym w praktyce krajowej. Przedstawienie historii takiej realizacji zakończonej sukcesem finansowym i moralnym może być pouczające dla wszystkich, którzy myślą o profesjonalnym programowaniu dla przemysłu produkującego oprogramowanie.

Centrum Naukowo-Produkcyjne Systemów Sterowania MERASTER w Katowicach jest producentem systemów komputerowych zgodnych z PDP-11 firmy Digital Equipment Corporation. Znaczna część produkcji jest eksportowana, głównie do ZSRR. W 1984 r. w produkcji znajdował się model MERA-60 z procesorem, będącym odpowiednikiem LSI-11. Komputer ten dawał niezbyt wielkie możliwości realizacji systemów wielodostępnych, głównie z uwagi na niewielki obszar pamięci adresowanej przez procesor, wynoszący 64 KB.

GENEZA SYSTEMU MASTER

Równocześnie odbiorcy krajowi i zagraniczni zaczęli wywierać coraz większy nacisk na dostarczenie narzędzi, umożliwiających realizację systemów wielodostępnych. W tej sytuacji, w CNPSS podjęto decyzję o opracowaniu w możliwie krótkim czasie sprzętu i oprogramowania, umożliwiającego realizację takich systemów. Rozwiązanie to uatrakcyjniające ofertę przedsiębiorstwa dotyczącą nowych wyrobów, miało być przydatne do rozszerzania już sprzedanych konfiguracji komputera MERA-60.

We wrześniu 1984 r. rozpoczęły się pierwsze prace koncepcyjne nad zestawem środków sprzętowo-programowych, który pozwoliłby na szersze użycie komputera MERA-60 w następujących dziedzinach zastosowań:
- obsługa masowa w bankach, kasach, urzędach, biurach podróży itp.,
- systemy automatycznego wspomagania procesu nauczania,
- systemy zbierania i wstępnego przetwarzania danych,
- systemy automatycznej obsługi kartotek w szpitalach, magazynach, bibliotekach itp.

W systemach tych zasadniczą rolę odgrywają takie cechy funkcjonalne, jak:
- wielodostęp, tzn. obsługa równoczesnej pracy wielu użytkowników na wielu terminalach monitorowych,
- wielozadaniowość, tzn. jednoczesna realizacja kilku różnych zadań przy użyciu różnych terminali,
- podział czasu pracy procesora między realizowane zadania,
- obsługa bazy danych systemu użytkowego,
- wymiana informacji między różnymi zadaniami i terminalami w systemie,
- wspólne korzystanie z niektórych urządzeń (np. drukarki) przez wiele zadań.

Przyjęto, że środkiem umożliwiającym zwiększenie możliwej do wykorzystania pamięci operacyjnej będzie **Półprzewodnikowa Pamięć Zewnętrzna (PPZ)**. Pojemność takiej pamięci powinna wynosić od 0,5 do 4 MB. Dolna granica zakładanej pojemności była umotywowana wstępnymi oszacowaniami potrzeb równolegle projektowanego oprogramowania, a górna – dostępną podczas projektowania bazą elementową.

Założono, że pamięć ta będzie pracować w dwóch trybach:
- jako emulator dysku SM 5400,
- jako rozszerzenie pamięci operacyjnej, tzn. urządzenie umożliwiające przełączenie obszaru pamięci widzianego przez procesor w określonym obszarze adresów pamięci operacyjnej.

Pierwszy tryb pracy miał umożliwić przyspieszenie pracy programów, które intensywnie komunikują się z nośnikiem pamięci masowej. Skopiowanie, na przykład, przetwarzanych przez program danych na dysk elektroniczny może kilkakrotnie przyspieszyć pracę programu.

W drugim trybie postanowiono w pamięci operacyjnej minikomputera wydzielić 8 KB w obszarze najwyższych adresów. Obszar ten stanowi „okno", które za pomocą odpowiednich mechanizmów sprzętowych (rejestry) i programowych jest odwzorowywane w sposób wzajemnie jednoznaczny na określoną stronę pamięci buforowej. Umożliwia to pracę na danej stronie pamięci buforowej w sposób analogiczny do adresowania pamięci operacyjnej. Istniejący mechanizm przełączania stron pamięci buforowej pozwala na pracę na różnych stronach, w różnych chwilach, z użyciem tych samych adresów z obszaru okna.

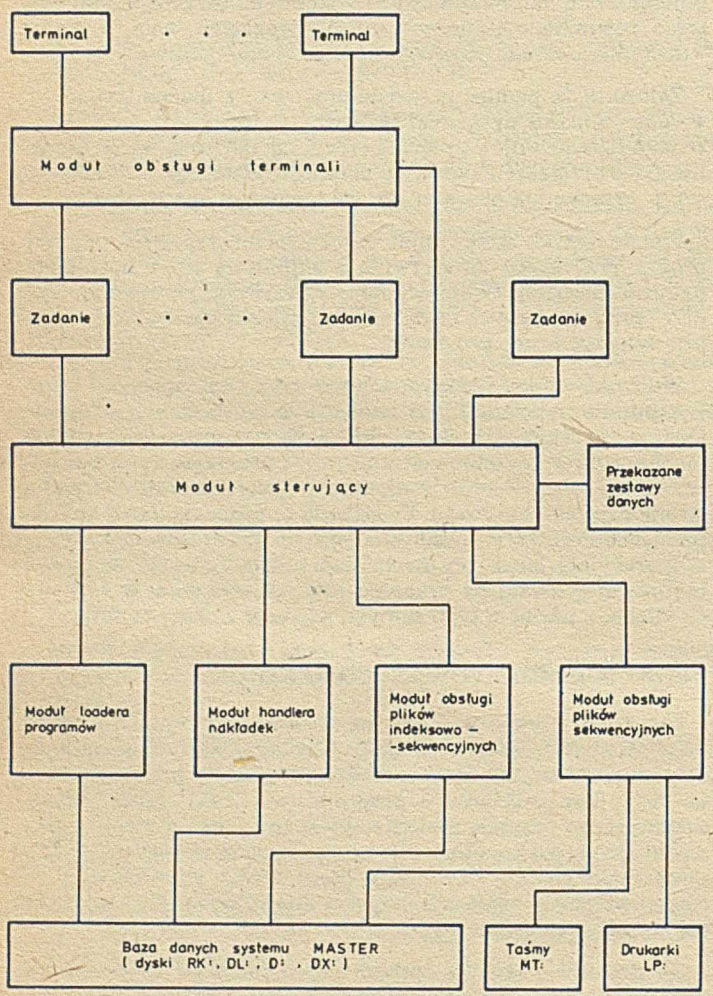
FUNKCJE PROJEKTOWANEGO SYSTEMU

Prace nad projektowanym rozwiązaniem sprzętowo-programowym rozpoczęto od dyskusji prowadzonej w gronie handlowców odpowiedzialnych za sprzedaż produktów przedsiębiorstwa, oraz konstruktorów i programistów. Taki skład zespołu definiującego przyszłą funkcjonalność produktu pozwolił uniknąć błędu często popełnianego przez twórców oprogramowania powtarzalnego, w którym duża część bardzo wyrafinowanych funkcji nie jest w ogóle wykorzystywana, odczuwa się natomiast brak funkcji elementarnych.

Na podstawie analizy potrzeb domniemanego użytkownika założono następujące funkcje projektowanego oprogramowania:
- **obsługa wielodostępu** – zapewnienie możliwości jednoczesnej pracy wielu użytkowników przy użyciu wielu terminali;
- **podział czasu** – sterowanie przydziałem kwantu czasu procesora dla zadania gotowego do wykonania;
- **sterowanie pamięcią buforową** – przydzielenie do zadań i zwalnianie stron pamięci buforowej oraz sterowanie przełączaniem stron aktualnych;
- **zarządzanie zadaniami** – sterowanie pracą wielu zadań aplikacyjnych związane z podziałem czasu i wielodostępem, zachowywanie i odtwarzanie stanów zadań;
- **zarządzanie bazą danych** – obsługa plików systemowych dla dostępu indeksowo-sekwencyjnego;

- **zarządzanie plikami sekwencyjnymi** – obsługa, przydział i zwalnianie urządzeń dla plików sekwencyjnych; urządzenia mogą być rezerwowane dla zadania lub wykorzystywane jednocześnie przez wiele zadań;
- **synchronizacja zadań** – obsługa przekazywania komunikatów i danych między zadaniami aplikacyjnymi;
- **obsługa przerwań i błędów** – przejmowanie niektórych przerwań od urządzeń oraz błędów programów aplikacyjnych w celu niedopuszczenia do załamania systemu;
- **ładowanie programów, nakładkowanie i łańcuchowanie** – sterowanie sposobem wykonania części zadań, ładowanie kolejnych nakładek do odpowiednich stron PPZ;
- **obsługa urządzeń zewnętrznych** – zarządzanie przydziałem i transmisją z i do urządzeń zewnętrznych systemu.

Następnie, już w gronie analityków systemu zdefiniowano strukturę systemu pod względem jego budowy. Pozwoliło to określić liczbę osób konieczną i wystarczającą do realizacji tematu. Projektowanemu systemowi operacyjnemu, który miał realizować powyższe funkcje, nadano nazwę **MASTER** (**M**ulti**A**ccess **T**imesharing System for **E**fficient **R**esources Usage). Jego strukturę przedstawiono na rys. 1.



Rys. 1. Struktura funkcjonalna systemu MASTER

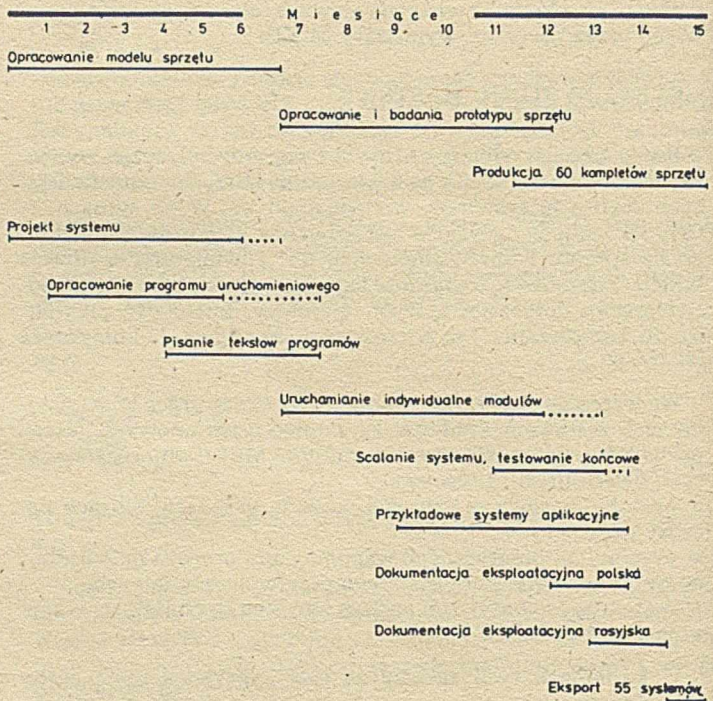
METODA REALIZACJI SYSTEMU

W chwili podjęcia decyzji o realizacji systemu MASTER i przystąpienia do jej wykonywania występowały liczne ograniczenia mające wpływ na sposób pracy. Podstawowym ograniczeniem była konieczność wykonania całej pracy w ciągu 15 miesięcy – przy czym rozpoczynano niemal z poziomu zerowego (koncepcja systemu programowego i koncepcja wspomagającego go sprzętu). Ten zdecydowanie krótki czas na wykonanie modułu sprzętowego (półprzewodnikowej pamięci zewnętrznej) i opartego na jego wykorzystaniu złożonego systemu programowego, wymusił przyjęcie specyficznego trybu realizacji systemu.

Pierwsza (niezbyt standardowa) decyzja dotyczyła maksymalnego zrównoleglenia wykonywanych zazwyczaj sekwencyjnie prac. Odnosiło się to zarówno do jednoczesnej realizacji sprzętu i oprogramowania, jak i wyprzedzającego rozpoczynania kolejnych etapów prac (przed zakończeniem poprzednich). Na rysunku 2 podano diagram czasowy głównych etapów pracy.

Równoległa realizacja prac wymagały przyjęcia rozwiązań organizacyjnych i technicznych nie stosowanych do tej pory przez nikogo z realizatorów. W celu przyspieszenia prac nad systemem oraz zwiększenia jego późniejszej niezawodności przyjęto w początkowym okresie opisane poniżej założenia i uzgodnienia (i konsekwentnie ich następnie przestrzegano).

- Jako technikę organizacji prac nad realizacją systemu przyjęto metodę Zespołu Głównego Programisty, tzn. taki sposób realizacji oprogramowania, w którym istnieje jedna osoba mająca pełną i całkowitą wiedzę o tworzonym systemie, druga mogąca po uzupełnieniu swojej wiedzy przejąć zadania głównego programisty oraz kilkanaście osób realizujących praktycznie niezależnie swoje fragmenty systemu. Taka organizacja pracy wymagała opracowania projektu systemu z największą dokładnością i przyjęcia ustalenia, że od pewnego momentu rozwoju oprogramowania jakiekolwiek zmiany (nawet drobne) nie mogą być do projektu wprowadzane.
- Przyjęto warstwowo-modułową strukturę tworzonego systemu – z dużą izolacją poszczególnych modułów systemu. Dawało to wprawdzie jako efekt uboczny pewną nieoptymalność systemu jako całości, lecz w istotnym stopniu zwiększało niezawodność systemu.
- Cała dokumentacja systemu (projektowana oraz w późniejszym etapie eksploatacyjna) była tworzona wyłącznie na nośniku magnetycznym.



Rys. 2. Przebieg prac nad systemem

- Proces uruchamiania poszczególnych modułów oraz scalania systemu był wspomagany przez specjalizowane narzędzia uruchomieniowe.
- Wprowadzono zindywidualizowany rozkład czasu pracy dla realizatorów systemu.

ZNACZENIE DOKUMENTATORA

Niestandardowy temat wymagał niestandardowego podejścia, dlatego też udało się dość łatwo przekonać przełożonych, że normalne miejsce pracy nie jest najlepsze do wstępnych dyskusji

nad koncepcją systemu. Dyskusja taka, trwająca tydzień, odbyła się w ośrodku wypoczynkowo-konferencyjnym. Każdy z ośmiu uczestników tego posiedzenia otrzymał mniej więcej dwa tygodnie wcześniej zadanie przygotowania i przedstawienia propozycji rozwiązania jednego z fragmentów systemu.

Po powrocie każdy wprowadził na nośnik maszynowy fragment wstępnej koncepcji systemu i pod koniec pierwszego miesiąca pracy dysponowano szesnastoma egzemplarzami zerowej wersji projektu. Zastosowanie formatera tekstów DOC60 miało kluczowe znaczenie w realizacji całego tematu. W ciągu pierwszych pięciu miesięcy pracy powstało pięć kolejnych wersji projektu technicznego systemu, coraz bardziej spójnych i szczegółowych.

Iteracyjny proces udoskonalania projektu, w którym brało udział szesnaście osób – późniejszych bezpośrednich realizatorów oprogramowania – był możliwy tylko dzięki temu, że kolejne wydania projektu były uzupełniane i drukowane na nowo po każdej rewizji powstającej w wyniku dwu lub trzydniowych dyskusji całego zespołu. Równocześnie projekt był pisany z myślą o wykorzystaniu jego fragmentów przy tworzeniu dokumentacji użytkowej.

SPECJALIZOWANE PROGRAMY URUCHOMIENIOWE

Pierwszą pracą ściśle programistyczną, podjętą natychmiast po określeniu struktur wewnętrznych systemu, było stworzenie specjalizowanych narzędzi uruchomieniowych. Normalne środki uruchomieniowe stosowane w systemie RT-60 nie mogły być zastosowane, choćby z uwagi na wymóg wielozadaniowości systemu.

Podstawowym narzędziem przy uruchamianiu programów systemu był specjalizowany program uruchomieniowy, który oprócz typowych możliwości udostępniał również wewnętrzne struktury systemowe. Umożliwiał on:

- sprawdzanie i modyfikowanie zawartości globalnych zmiennych systemowych,
- rejestrowanie na bieżąco przejść przez ustalone punkty kontrolne w testowanych programach,
- rejestrowanie wszelkich zmian globalnych zmiennych systemowych (czas i moduł dokonujący zmiany, zawartość pierwotna i zmodyfikowana),
- odtwarzanie w dowolnym momencie końcowego fragmentu historii systemu, na które składały się zarejestrowane przejścia przez punkty kontrolne i zmiany globalnych zmiennych systemowych.

Poszczególne moduły programowe współpracowały z programem uruchomieniowym za pośrednictwem zestawu mikroinstrukcji. Podczas uruchamiania i testowania poszczególnych modułów były używane wybrane możliwości programu uruchomieniowego. Pozwalał on selektywnie wybierać stosowane w danej chwili mechanizmy, a po wystarczającym przetestowaniu modułu – zrezygnować z nich całkowicie.

ŚRODKI ORGANIZACYJNE

W celu przyspieszenia realizacji prac nad oprogramowaniem systemu zastosowano również niestandardowe rozwiązania organizacyjne. Najistotniejszym z nich była rezygnacja ze stałych godzin wykonywania prac przez realizatorów systemu. Pozwoliło to na pracę kilkunastoosobowego zespołu na jedynym egzemplarzu sprzętu, wykorzystywanym praktycznie przez 24 godziny na dobę i przez siedem dni w tygodniu.

Istnienie kilku wydań „Projektu Technicznego" w nakładzie po kilkanaście egzemplarzy powodowało konieczność stworzenia stanowiska Sekretarza projektu, tj. osoby odpowiedzialnej za wycofanie z obiegu egzemplarzy nieaktualnych i przechowywanie wszystkich kolejnych wydań.

Dopuszczono wprowadzenie tylko niezbędnych poprawek lub zmian w stosunku do ostatecznej wersji projektu. Ustalanie niezbędności było dokonywane w trybie dwu- lub wielostronnym, w zależności od tego, ilu modułów dotyczyła poprawka, natomiast jakiekolwiek najmniejsze odstępstwo w stosunku do projektu wymagało zgody Głównego Programisty.

Pomysły gruntownych zmian, nasuwające się realizatorom w trakcie pisania i uruchamiania systemu, były notowane w aneksie do projektu, dzięki czemu wraz z realizacją pierwszej wersji systemu powstał projekt techniczny wersji drugiej.

DOŚWIADCZENIA Z REALIZACJI SYSTEMU

Główną przyczyną powodzenia projektu była oprócz rzetelnego przygotowania założeń i konsekwentnej ich realizacji – są to bowiem podstawy profesjonalnej działalności – rezygnacja zespołu z dążenia do ideału. Błędem z punktu widzenia programowania systemu przemysłowego byłoby wprowadzanie nawet najdoskonalszych uzupełnień kosztem efektu końcowego. Efektem takich ulepszeń wprowadzanych w trakcie projektowania mogłoby być to, że zespół zrealizowałby, być może:
Najlepszy Na Świecie System Na Wszystkie Okazje
(w stadium realizacji), zamiast dobrego ukończonego systemu.

Kolejnym powodem sukcesu było wyzwanie postawione przez skalę trudności systemu. Należy przypuszczać, że realizacja problemu o zbliżonej złożoności, ale rutynowego, nie wyzwoliłaby wśród wykonawców takich rezerw.

Innym istotnym czynnikiem sprzyjającym powodzeniu projektu było to, że twórcy systemu stali się jego pierwszymi użytkownikami. Jeszcze podczas realizowania systemu, ci z jego realizatorów, którzy zakończyli swój udział w jego powstawaniu, rozpoczęli natychmiast tworzenie sporego systemu aplikacyjnego, na zamówienie zewnętrznego klienta. Takie podejście, jakkolwiek związane z ryzykiem niewykonania dwóch prac naraz, pozwoliło lepiej zrealizować oba tematy.

Kolejnym pozytywnym doświadczeniem jest metoda eliminacji efektów wszystkich praw Murphy'ego, polegająca na uprawianym do znudzenia składowaniu najświeższych efektów pracy na nośnikach zapasowych, z zachowaniem ostatnich dwóch (ale nie więcej) kopii. W praktyce stosowana metoda archiwowania pozwoliła uniknąć większych problemów w trakcie realizowania systemu.

EFEKTY

Prace nad realizacją systemu MASTER i wspomagającego go sprzętu, przyniosły zarówno efekty materialne, jak i należące do sfery niematerialnej. Podstawowym efektem materialnym było sprzedanie (w znacznej części na eksport) kilkudziesięciu systemów komputerowych z nowo opracowanym sprzętem i oprogramowaniem.

Za drugi wynik z grupy efektów materialnych należy uznać znaczące rozszerzenie konfiguracji i możliwości funkcjonalnych systemu komputerowego produkowanego już od kilku lat. Uwzględniając fakt dostarczenia w tym czasie odbiorcom kilku tysięcy komputerów – umożliwienie rozbudowy tych już eksploatowanych systemów daje ich użytkownikom znaczne potencjalne korzyści.

Niemniej ważnym niż poprzednie wynikiem jest efekt całkowicie niezamierzony, jakim są doświadczenia i umiejętności wykonawców wyniesione z prac nad wytworzonym systemem. Wymienić trzeba tutaj zarówno doświadczenie i umiejętności czysto programistyczne, wynikające z realizacji złożonego systemu w pełnym cyklu od koncepcji, aż do wdrożenia i eksploatacji, jak i wnioski i doświadczenia praktyczne z dziedziny organizacji prac nad złożonym systemem programowania.

Doświadczenia te zostały zresztą docenione w postaci nagrody I stopnia dla CNPSS MERASTER, na Ogólnopolskich Targach Oprogramowania SOFTARG'86, oraz I nagrody Specjalnej Ministra, Kierownika Urzędu Postępu Naukowo-Technicznego i Wdrożeń dla zespołu autorów oprogramowania. Innym potwierdzeniem wzrostu poziomu umiejętności autorów jest zgłoszenie zastrzeżenia patentowego przez zespół konstruktorów Półprzewodnikowej Pamięci Zewnętrznej.

LITERATURA

[1] Korniak T., Fuglewicz P.: Wielodostępny system operacyjny z podziałem czasu MASTER – budowa i zastosowania. Biuletyn Techniczno-Informacyjny MERA, nr 7 (289), 1986

[2] Dokumentacja firmowa systemu MASTER. Centrum Naukowo-Produkcyjne Systemów Sterowania MERASTER, Katowice, 1986.

JAN ZABRODZKI
Instytut Informatyki
Politechnika Warszawska

POLSKIE TOWARZYSTWO INFORMATYCZNE

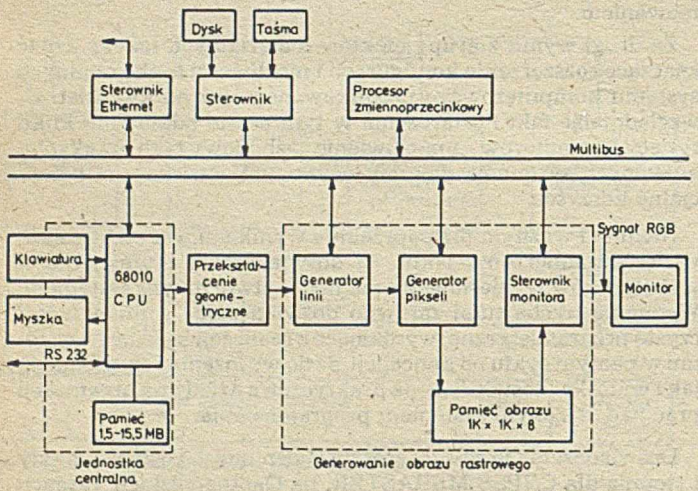
Cyfrowe generowanie obrazów (2)

W drugiej części artykułu omówiono stacje graficzne oraz tzw. metodę śledzenia promieni.

STACJE GRAFICZNE

Do niedawna jedyną możliwość korzystania z techniki graficznej wysokiej jakości dawały duże systemy komputerowe połączone z drogimi terminalami graficznymi. Komputer obciążony był zarówno rozwiązywaniem określonego zadania, jak i przetwarzaniem informacji w celu utworzenia obrazu wyświetlanego na terminalu. W miarę postępu technologii pojawiła się tendencja do rozszerzania funkcji spełnianych przez terminal o zadania związane z przygotowaniem wyświetlanego obrazu. Tendencja ta doprowadziła do powstania stacji graficznych. Są to urządzenia, które współpracując z dużym komputerem przejmują wszystkie funkcje graficzne. Równocześnie stacje takie często są wyposażone w mechanizmy konieczne do pracy interakcyjnej i są zdolne do samodzielnej pracy. Niżej podana jest charakterystyka stacji graficznej IRIS 2400 firmy Silicon Graphics.

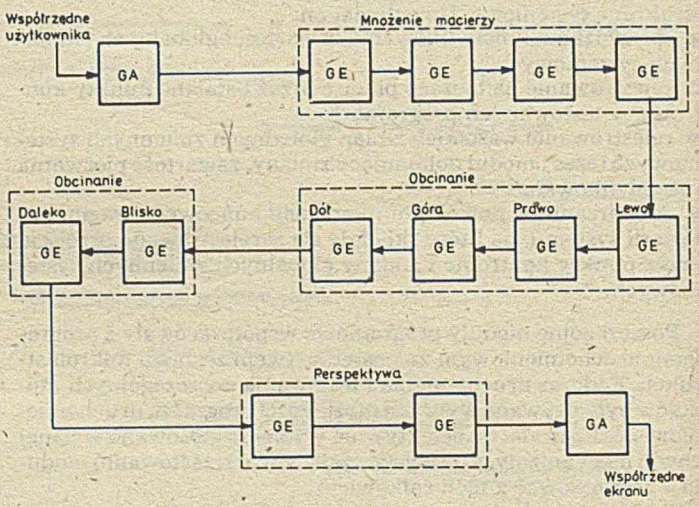
Jednostka centralna stacji jest zbudowana z wykorzystaniem mikroprocesora 68010. Stacja jest wyposażona w pamięć RAM o pojemności od 1,5 do 15,5 MB i wykorzystuje system operacyjny UNIX. Pojemność pamięci dyskowej typu Winchester wynosi od 72 MB do 440 MB. Rozdzielczość 19-calowego monitora kolorowego wynosi 1024 × 768 pikseli (przy wyświetlaniu z częstotliwością 60 Hz). Stacja może pracować niezależnie bądź też może współpracować z siecią Ethernet. Dostępna jest też magistrala Multibus. Stacja wyposażona jest w system okien.



Rys. 1. Stacja graficzna IRIS 2400

Na rys. 1 przedstawiono schemat blokowy stacji. W systemie wykorzystywana jest idea przetwarzania potokowego, zarówno na poziomie trzech podstawowych bloków jednostki centralnej, bloku przekształceń geometrycznych i bloku generacji obrazu rastrowego, jak też i wewnątrz tych bloków.

Najciekawszym elementem stacji jest blok przekształceń geometrycznych zbudowany na podstawie dwu typów układów VLSI: układzie przekształceń geometrycznych (ang. geometry engine – GE) oraz układzie przyspieszającym (ang. geometry accelerator – GA). Sposób połączenia układu w strukturze potokowej wyjaśnia rys. 2. W skład układu GE wchodzą m. in. cztery 32-bitowe, zmiennoprzecinkowe jednostki arytmetyczne oraz pamięć sterująca. W strukturze potokowej wykorzystuje się 10 lub 12 takich układów. Każdy z nich spełnia jedną z trzech funkcji określanych programowo. Pierwsze cztery układy wykonują operacje na macierzach współrzędnych jednorodnych (4 × 4) dla uzyskania obrotu, przesunięcia i skalowania. Następne cztery lub sześć układów GP ma zadanie realizacji funkcji obcinania dla określenia części obrazu należącej do ostrosłupa widzenia (z uwzględnieniem lub bez uwzględnienia głębokości obrazu). Ostatnie trzy układy GP wykonują operacje dzielenia dla uzyskania perspektywy i przejścia na dwuwymiarowe współrzędne ekranu. Układy przyspieszające GA mają do spełnienia dwa zadania: buforowanie (na zasadzie FIFO, konieczne ze względu na różnice szybkości pracy poszczególnych bloków) i konwersję na lub z postaci zmiennoprzecinkowej, wykorzystywanej w układach GE. Efektem zastosowanych rozwiązań jest szybkość przekształceń geometrycznych rzędu 65 000 transformacji na sekundę.



Rys. 2. Struktura potokowa bloku przekształceń geometrycznych

Współrzędne punktów uzyskiwane na wyjściu bloku przekształceń geometrycznych są wykorzystywane w bloku generowania obrazu rastrowego do utworzenia obrazu w pamięci obrazu. W pierwszej kolejności 16-bitowy procesor segmentowy generuje linie i wielokąty. Z kolei blok generowania zawartości pamięci obrazu realizuje algorytmy związane z wypełnianiem wielokątów. Wypełnianie wielokątów wypukłych odbywa się z szybkością 44 mln pikseli/s. Realizowane są tu również algorytmy cieniowania (według Gouraud) oraz określenia intensywności linii i punktów w funkcji odległości od obserwatora. Operacje te mogą być realizowane z szybkością do 3 mln pikseli/s. Układ sterownika monitora zapewnia generację sygnałów R, G, B.

W stacji IRIS 2400 i innych stacjach firmy Silicon Graphics, konstruktorzy zdecydowali się na wykorzystanie dwóch specjalizowanych układów VLSI, własnej konstrukcji. Inne firmy na ogół dążą do korzystania z układów dostępnych handlowo. W tabeli podano podstawowe parametry kilku wybranych stacji graficznych.

Większość stacji pozwala wyświetlać obrazy z rozdzielczością 1 K × 1 K i wykonywać od 0,4 do 1,5 mln operacji/s. Stacje te są przystosowane do pracy w sieciach lokalnych. Odpowiednią moc obliczeniową uzyskuje się jednym z trzech sposobów: stosowania wielomikroprocesorowych systemów 16-bitowych, wykorzystywania mikroprocesorów 32-bitowych oraz konstruowania procesorów segmentowych. Standardem jest stosowanie systemu operacyjnego Unix. Większość systemów wykorzystuje własne magistrale wewnętrzne, jednak prawie wszystkie stacje mają sprzężenie z magistralą Multibus. Również prawie wszystkie stacje są wyposażone w system okien.

METODA ŚLEDZENIA PROMIENI

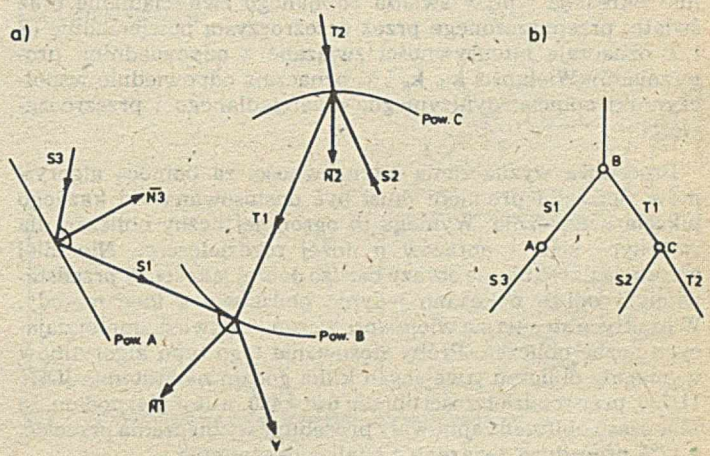
Dążąc do uzyskania obrazów jak najbardziej zbliżonych do rzeczywistych obiektów, opracowano wiele różnych metod rozwiązywania poszczególnych problemów, takich jak eliminowanie linii i powierzchni niewidocznych, symulowanie efektów oświetlenia, odbić, cieni, zmiany jasności w funkcji odległości itp. Dla realizacji niektórych z tych metod opracowano specjalne rozwiązania sprzętowe. Jednak w wielu wypadkach urządzenia dobrze nadające się do realizacji jednego rodzaju algorytmu, gorzej radzą sobie z innymi algorytmami.

Metodą w pewnym sensie uniwersalną jest metoda śledzenia promieni. Jest to jednak metoda wymagająca bardzo dużej liczby obliczeń i jak dotychczas jej realizacja jest możliwa za pomocą dużych komputerów. Metodę tę cechuje pewna regularność obliczeń i możliwości zrównoleglania obliczeń, co rokuje nadzieje na opracowanie specjalizowanych układów VLSI bądź wykorzystanie innych urządzeń do przetwarzania równoległego.

Idea metody jest następująca. Analizuje się bieg każdego promienia, który przechodząc przez piksel obrazu dociera do oka obserwatora. Dla każdego takiego promienia wyznacza się przecięcie ze wszystkimi obiektami oglądanej sceny. Punkt przecięcia leżący najbliżej oka należy do obiektu widzialnego.

Po określeniu dla danego promienia powierzchni widocznej generowany jest drugi promień, który podlega odbiciu od tej powierzchni zgodnie z prawami optyki. W wypadku, gdy odbity promień przecina inne obiekty sceny, to pierwszy taki obiekt jest widziany jako obraz odbity. Bieg promienia jest śledzony dalej tak długo, aż przestanie przecinać obiekty należące do sceny. Teraz możliwe jest ustalenie intensywności i koloru dla rozpatrywanego piksela, przez wsteczne śledzenie promienia i rekurencyjne stosowanie w każdym punkcie odbicia przyjętego modelu oświetlenia. Uwzględnia się przy tym funkcję tłumienia zależną od odległości między kolejnymi punktami odbicia. Po zakończeniu procesu cofania uzyskuje się wartość intensywności lub kolor dla rozpatrywanego piksela, z uwzględnieniem wpływu oświetlenia pochodzącego od wszystkich istotnych obiektów należących do analizowanej sceny.

W wypadku gdy w scenie występują obiekty przezroczyste, w każdym punkcie przecięcia śledzonego promienia z powierzchnią generuje się dwa promienie: jeden odbity i jeden załamany. Kierunek promienia załamanego określa się zgodnie z prawem Snelliusa. Również i dla tego promienia uwzględnia się efekty tłumienia w funkcji odległości i współczynnika przezroczystości. Zasadę generowania promieni ilustruje rys. 3a. Natomiast na rys. 3b pokazano fragment drzewa, które jest tworzone dla każdego piksela. Każdy węzeł drzewa odpowiada punktowi przecięcia promienia z powierzchnią. Promienie S_i są promieniami odbijanymi zwierciadlanie, natomiast promienie T_i reprezentują promienie załamane. W ogólnym wypadku gałęzie kończą się, gdy ich udział w końcowym bilansie intensywności piksela staje się na tyle mały, że można go pominąć (albo też wyczerpuje się dostępna pamięć).



Rys. 3. Promienie generowane dla jednego piksela i odpowiednie drzewo promieni; powierzchnia A jest nieprzezroczysta

W wypadku, gdy uwzględnia się problem cieni rzucanych przez obiekty, w każdym punkcie przecięcia promienia z powierzchnią generuje się kolejne promienie skierowane do poszczególnych źródeł światła. Jeśli promień taki przetnie jakiś obiekt zanim dotrze do źródła światła, to punkt, w którym ten promień

Parametry wybranych stacji graficznych

Producent	Możliwości obliczeniowe						Grafika		
	Procesor	Pamięć podręczna (w bajtach)	Pamięć RAM (pojemność w bajtach)	Pamięć wirtualna (pojemność w bajtach)	Procesor zmienno-przecinkowy	System operacyjny	Rozdzielczość ekranu	Szybkość rysowania (piksele/s)	Kolor, liczba bitów na piksel
Apollo Computers	Segmentowy	4 K na rozkazy 16 K na dane	4 M	256 M	Własny	Aegls Unix	1024 × 1024	1,5 M	8 lub 24
DEC	VAX-780	–	9 M	4 G	Własny	VMS UNIX 4.2	1280 × 1024 (terminal Tektronix 6125)	500 K	2 lub 8
Gould Inc.	32-bitowy własny	32 K na rozkazy, 32 K na dane	16 M	16 M	Własny	Unix 4.2	1280 × 1024	1 M	8, 16 lub 24
Hewlett-Packard	68010 lub 6820	16 K	7,5 M	4 G	68881	HP-UX	1024 × 768 (zewnętrzny monitor)	2,5 M	–
Jupiter Systems	68010	–	4 M	4 M	Format IEEE	Unix System III	1280 × 1024	30 M	4 lub 8
Mossic Technology	32032	16 K	8 M	16 M	32082	Unix 4.2	1280 × 1024	1,6 M	4, 8, 16 lub 32
Ridge Computers	32-bitowy RISC	–	8 M	4 G	Tak (w CPU)	Unix System V	1024 × 800	37 M	Czarno-biały
Silicon Graphics	68010	–	15 M	16 M	W sterowniku graficznym	Unix System V	1024 × 1024	3 M	8 lub 32
Sun Microsystems	68010 lub 68020	–	8 M	16 M	68881	Unix 4.2	1152 × 900	1,5 M	8
Tektronix	32016 lub 32032	–	10 M	16 M	32081	Unix 4.2	1024 × 768	2,7 M	4 lub 8

został wytworzony, znajduje się w obszarze cienia związanego z danym źródłem światła. Możliwe jest również uwzględnianie przypadku obiektów przezroczystych znajdujących się między analizowanym punktem a źródłem światła. Uwzględnianie wpływu cieni na końcową intensywność piksela jest możliwe przez przyjęcie odpowiedniego modelu oświetlenia wykorzystywanego sekwencyjnie przy analizie każdego węzła drzewa.

W artykule [14] przyjęty jest następujący model oświetlenia:

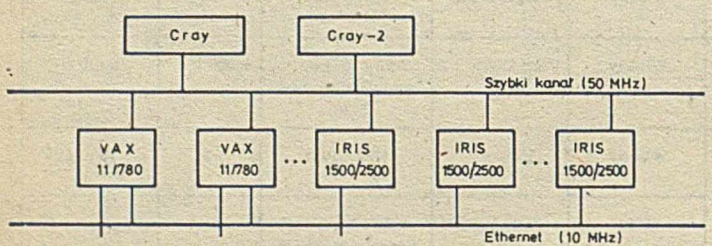
$$I = I_a + k_d \sum_{j=1}^{j=1_s} (\bar{N} \cdot \bar{L}_j) + k_s S + k_t T$$

Intensywność I w danym punkcie powierzchni jest sumą czterech składników. Czynnik $\mathbf{I_a}$ określa wpływ światła otoczenia. Drugi czynnik pozwala uwzględnić wpływ światła odbijanego, którego intensywność zgodnie z prawem Lamberta jest proporcjonalna do cosinusa kąta między normalną do powierzchni ($\bar{N}$) a wektorem kierunku źródła światła ($\bar{L}j$). Pozostałe dwa czynniki określają wpływ światła odbijanego zwierciadlanie oraz światła przepuszczonego przez przezroczystą powierzchnię (**S** i **T** oznaczają intensywności związane z odpowiednimi promieniami). Wielkości $\mathbf{k_d}$, $\mathbf{k_s}$ i $\mathbf{k_t}$ oznaczają odpowiednio współczynniki odbicia: dyfuzyjnego, zwierciadlanego i przezroczystości.

Procedura wyznaczania intensywności za pomocą algorytmów śledzenia promieni musi być dostosowana dla każdego piksela niezależnie. Wymaga to ogromnej liczby obliczeń dla złożonych scen i obrazów o dużej rozdzielczości. Niemniej w efekcie uzyskuje się obrazy bardzo dobrej jakości. W przedstawionym opisie pokazano jedynie podstawową ideę metody. W praktyce stosuje się wiele uproszczeń i ułatwień zmniejszających liczbę obliczeń. Próby stosowania tego typu algorytmów wymagały obliczeń trwających kilka godzin na systemie VAX 11/780, przy rozdzielczości obrazu 640 × 480. Autor [14] podaje, że 75% czasu obliczeń zajmowały procedury wyznaczania przecięć, a 12% procedury związane z analizą oświetlenia.

* * *

W artykule przedstawiono wybrane problemy związane z konstruowaniem urządzeń do cyfrowego generowania obrazów i ich wyświetlania. Przegląd ten pokazuje tendencje rozwoju tego typu urządzeń. Specjalizowane układy o coraz większym stopniu scalenia umożliwiają budowę urządzeń graficznych o coraz lepszych parametrach i coraz powszechniej dostępnych. Jednak, jak pokazuje choćby przykład metody śledzenia promieni, nie wszystkie problemy grafiki komputerowej dają się rozwiązywać za pomocą dostępnych obecnie specjalizowanych urządzeń graficznych. Dla niektórych problemów nadal jedyną możliwość rozwiązania stanowią złożone systemy superkomputerowe.



Rys. 4. System ze stacjami graficznymi

W ośrodku firmy Digital Production [3] zainstalowana jest konfiguracja wykorzystywana do generowania obrazów animowanych i specjalnych efektów o bardzo wysokiej jakości. Dla zobrazowania złożoności problemu warto przytoczyć kilka liczb. Konfiguracja, w skład której wchodzą: Cray X-MP/2230 i VAX 11/782, potrzebuje od 3 s do 10 h pracy dla wytworzenia 1 s filmu animowanego. Daje to w rezultacie możliwość wytworzenia około 144 minut filmu rocznie.

Przykład innego systemu, wykorzystującego stacje graficzne, przedstawiono na rys. 4 [6]. Jest to system zainstalowany w jednym z ośrodków NASA do celów symulacji efektów aerodynamicznych. W systemie tym przewidziano wykorzystanie 25 opisanych wyżej stacji firmy Silicon Graphics.

LITERATURA

[1] Berk T.: Realistic image generation techniques and issus. Fourth Symposium on Microcomputer and Microprocessor Applications, Budapeszt, 1985, str. 1–13
[2] Clark J.: The geometry engine – a VLSI geome try system for graphics, Computer Graphics, pp. 127-133, July 1982
[3] Demos G. et al.: Digital scene simulation – the synergy of computer technology and human creativity, Proc. of the IEEE, pp. 22–31. January 1984
[4] Foley J. D., van Dam A.: Fundamentals of interactive computer graphics. Addison-Wesley, Reading (MA), 1982
[5] Greenberg D. et al.: The computer image – applications of computer graphics. Addison-Wesley, Reading (MA), 1982
[6] Gustafson P.: Graphics systems deliver upfront performance. Computer Design, pp. 61–65, 15 July 1985
[7] Leibson S. H.: Graphics-controller ICs, EDN, pp. 104–118, 6 February 1986
[8] Masewicz T.: Telewizja dla praktyków, WKiŁ, Warszawa 1982
[9] Newman W. M., Sproul R. F: Principles of interactive computer graphics, McGraw-Hill, New York, 1979
[10] Panasuk C.: Focus on graphics workstations. Electronics Design, pp. 157–164, 22 August 1985
[11] Price S. M.: COMOS 256-K bit video RAM with wide two-way bus, picks up speed, drops power. Electronics Design, pp. 171–177, 19 September 1985
[12] Whitton M. C.: Memory design for raster graphics displays. Computer Graphics and Applications, pp. 48–64, March 1984
[13] Wiliams T.: Graphics processing migrates from host to workstation. Computer Design, pp. 49–57, 15 July, 1985
[14] Witted T.: An improved illumination model for shaded display. Communication of the ACM, pp. 343–349, June 1980
[15] Wright M.: Color monitors. EDN, pp. 107–122, 31 May, 1984.

Ceny ogłoszeń

Od 1 lipca 1987 r. obowiązują następujące ceny materiałów reklamowych publikowanych na łamach INFORMATYKI:

Ogłoszenia

- ogłoszenia czarno-białe, artykuły reklamowe i informacje naukowo-techniczne (biuletyny) zależnie od objętości: cała strona – 50 tys., 3/4 str. – 45 tys., 2/3 str. – 40 tys., 1/2 str. – 35 tys., 1/3 str. – 30 tys., 1/4 str. – 25 tys., 1/8 str. – 20 tys., poniżej 1/8 str. – 200 zł za 1 cm²,
- ogłoszenia drobne (zależnie od liczby słów) jedno słowo – 50 zł

Dodatki do ceny podstawowej:

- za każdy dodatkowy kolor + 30%,
- za każdy specjalny kolor (nie wynikający z podstawowych kolorów) + 30%,
- za pełny kolor (grafika wielobarwna, zdjęcia kolorowe) + 120%,
- za zamieszczenie ogłoszenia na I lub IV stronie okładki + 100%
- za zamieszczenie ogłoszenia na II i III stronie okładki + 50%

Zniżki

dotyczą ogłoszeń – całkowitych powtórzeń

- za ogłoszenia 3–5-krotne – 10%
- za ogłoszenia 6–10-krotne – 20%
- za ogłoszenia 11-krotne i powyżej – 30%
- za artykuły i wkładki reklamowe wykonane przez zleceniodawcę – 40%
- za biuletyny i bloki reklamowe – 60%

W uzasadnionych wypadkach stosuje się zniżki specjalne dla ogłoszeń nie będących powtórzeniami – za zgodą Dyrektora – Naczelnego Redaktora Wydawnictwa NOT SIGMA. Ponadto Biuro Ogłoszeń świadczy usługi w zakresie wykonywania zdjęć czarno-białych i barwnych oraz nadbitek wkładek reklamowych.

Ceny wkładek

- wkładka 2 str. o formacie A4
  nakład do 500 egz. 20 tys. zł
  nakład 500–1000 egz. 35 tys. zł
- wkładka 4 str. o formacie A4
  nakład 500 egz. 40 tys. zł
  nakład 500–1000 egz. 70 tys. zł

Przy wkładkach o nakładzie powyżej 1000 egz. stosuje się wielokrotność powyższych cen, a dodatki za kolory obliczą się jak dla ogłoszeń.

Ogłoszenia przyjmowane się przez:
Dział Ogłoszeń i Reklamy WCiKT NOT SIGMA
ul. Świętojerska 5/7, 00–236 Warszawa
adres do korespondencji: skrytka pocztowa 1004, 00–950 Warszawa
telefony: 31-93-65 lub 31-22-21 w. 196 i 291

WOJCIECH MOKRZYCKI
Instytut Podstaw Informatyki
Polska Akademia Nauk
Warszawa

Techniki obrazowania informacji o trójwymiarowych zdyskretyzowanych scenach

Opanowanie technik wprowadzania i przetwarzania obrazów przez komputer w stopniu mogącym mieć praktyczne zastosowanie zostało bardzo szybko wykorzystane w wielu dziedzinach do pozyskiwania i przetwarzania informacji o wewnętrznej budowie przestrzennych obiektów i ich usytuowaniu na trójwymiarowej (3W) scenie, tj. wydzielonym obszarze przestrzeni 3W.

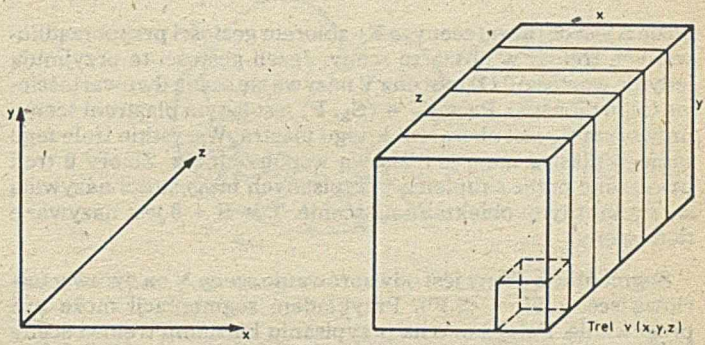
Istnieją różne techniki obrazowania informacji o trójwymiarowych obiektach, jak np. powszechnie stosowane techniki filmowa i telewizyjna lub grafika komputerowa, umożliwiająca zarówno komputerowe przetwarzanie, jak i prezentowanie prawie naturalnych obrazów. Gdy jednak wymagana jest informacja nie tylko na temat wzajemnego usytuowania obiektów na scenie, ale również wewnętrznej budowy obiektów, to wymienione techniki, nawet „konwencjonalna" generacyjna grafika komputerowa, są nie w pełni przydatne. W diagnostyce medycznej lub defektoskopii materiałowej, gdzie jest potrzebna informacja o wewnętrznej strukturze obiektów, np. organów ciała ludzkiego, wykorzystuje się powstałą niedawno trójwymiarową tomografię komputerową.

Skaner (omiatacz) tomografu wytwarza dużą liczbę dyskretnych obrazów, tzw. obrazów plastrowych (ang. slice image), wykonując kolejne równoległe „przekroje" trójwymiarowej sceny, np. za pomocą wiązki promieni rentgenowskich. Obrazy te zawierają informację o wewnętrznej budowie obiektów sceny. Stos obrazów plastrowych, poukładanych w kolejności ich wytworzenia tworzy dyskretną 3W scenę.

Dyskretne dwuwymiarowe (2W) obrazy są dwuwymiarowymi matrycami, zwykle kwadratowymi; inaczej mówiąc są szykami dwuwymiarowych elementów obrazu, zwanych dwelami lub komórkami (ang. pixel), ze sprzężonym z każdym dwelem wektorem, określającym pewne jego właściwości (np. przenikalność, szarość, kolor, intensywność itp.). Naturalną i bezpośrednią drogą obrazowania takich dwuwymiarowych dyskretnych obrazów jest powiązanie właściwości dwela w obrazie z intensywnością lub (i) kolorem świecenia odpowiedniego punktu na ekranie monitora. W przypadku scen trójwymiarowych nie ma niestety, takich prostych i bezpośrednich metod obrazowania informacji trójwymiarowej na dwuwymiarowym ekranie.

Obrazowanie 3W informacji za pomocą obrazów plastrowych jest nieadekwatne do niektórych potrzeb. Trzeba więc napierw wydobyć z dyskretnej trójwymiarowej sceny interesującą informację, a następnie przedstawić ją na płaskim ekranie. W obrazowaniu informacji o trójwymiarowej dyskretnej scenie istotną rolę odgrywają zarówno sposoby wydobycia i obrazowania informacji o scenie jak i wyobraźnia obserwatora, tj. jego możliwości percepcyjne.

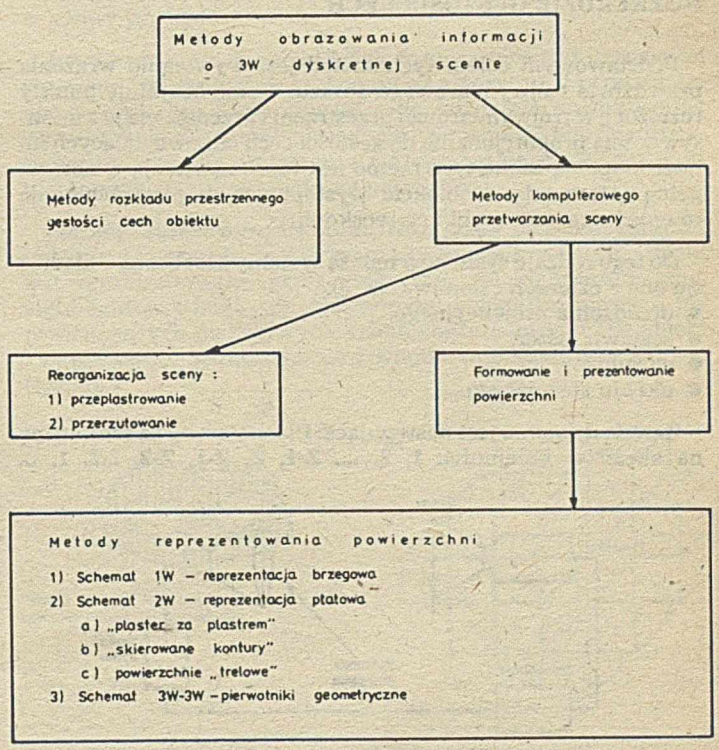
Zakładając, że uzyskany w trakcie omiatania sceny w pewnej płaszczyźnie obraz nie zmienia się aż do następnej (równoległej) płaszczyzny omiotu, każdy taki obraz jest dwuwymiarową matrycą identycznych prostokątnych objętościowych elementów, z przypisanym każdemu z nich wektorem określającym pewne jego cechy. Zbiór dwuwymiarowych obrazów, odnoszonych do trójwymiarowej sceny (tj. traktowanych jako obrazy plastrowe tej sceny), jest więc trójwymiarowym szykiem, tj. stosem plastrów złożonych z prostokątnych objętościowych elementów (rys. 1), z wartościami reprezentującymi właściwości każdego elementu. Analogicznie do dweli w dyskretnych dwuwymiarowych obrazach, trójwymiarowe elementy sceny są nazywane trelami (ang. voxel). Wartość sprzężoną z trelem nazywa się jego gęstością (właściwości).



Rys. 1. Trójwymiarowa scena jako stos „obrazów plastrowych"

Znane są dwie koncepcyjnie odmienne metody wydobycia i przedstawienia informacji o dyskretnej scenie (rys. 2):

- informacja o scenie jest otrzymywana z rozkładu cech obiektu w trójwymiarowej przestrzeni,



Rys. 2. Metody obrazowania dyskretnej informacji o trójwymiarowej scenie

- z obrazów plastrowych sceny jest wydzialana informacja o powierzchniach ograniczających obiekty; następnie te powierzchnie są obrazowane na płaskim ekranie monitora.

Wybór metody zależy od zastosowania, a jej efektywność od następujących wymagań użytkowych:

- w jakim stopniu dwuwymiarowy obraz opisuje informację trójwymiarową,
- czy zastosowana metoda umożliwia proste manipulowanie sceną oraz obiektami na scenie.

Drugie wymaganie jest ważne zwłaszcza wówczas, gdy trzeba obrazować izolowane struktury (kształty), wzajemnie się przesłaniające.

W każdej z metod wykorzystuje się model (matematyczny opis) dyskretnej sceny i obiektu. W modelu tym scena stanowi prostokątny równoległościan $S = X \times Y \times Z$, zwany obszarem sceny, który jest podzielony za pomocą zbioru równoległych płaszczyzn wzajemnie do siebie prostopadłych na identyczne równoległościany, nazywane trelami. Równoległościan ten umieszcza się w prostokątnym układzie współrzędnych **Oxyz** w taki sposób, aby dowolny trel **V** mógł być opisany za pomocą trójki liczb całkowitych **(x,y,z)**. Dyskretną trójwymiarową sceną **V** jest para:

$$V = (S,F)$$

gdzie **S** jest obrazem sceny, a **F** – zbiorem gęstości przyporządkowanych trelom w obszarze sceny. Jeżeli gęstości te przyjmują jedynie wartości **0** i **1**, to scena **V** nazywa się sceną dwuwartościową (albo binarną). Para $V_k = \langle S_k, F \rangle$ jest **k**-tym plastrem sceny, przy czym S_k jest obszarem **k**-tego plastra. Wszystkie trele tego samego plastra mają jednakową współrzędną **z**. Zbiory **0** treli utworzone przez skupienia przypisanych im gęstości nazywają się dyskretnymi obiektami na scenie. $T = S - 0$ jest nazywane tłem sceny.

Segmentacją sceny jest odwzorowanie sceny **V** na dwuwartościową scenę $V' = (S,F')$. Przykładem segmentacji może być progowanie. Polega ono na przypisaniu każdemu trelowi sceny **S** gęstości równej **1**, gdy jego rzeczywista gęstość na zadany poziom lub wartości gęstości równej **0**, gdy poziom rzeczywisty gęstości jest niższy od zadanego. Segmentacja służy zwykle do identyfikowania interesujących obiektów na scenie. Po wykonaniu progowania obiekt stanowią te trele, których gęstości cech mają wartość **1**.

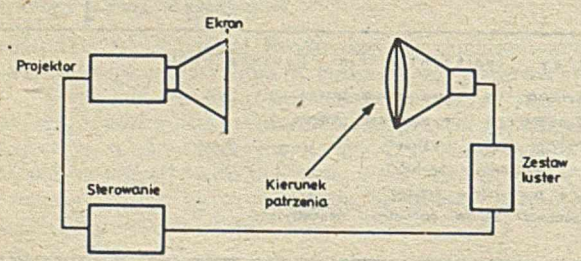
METODY OBRAZOWANIA OPARTE NA PRZESTRZENNYM ROZKŁADZIE GĘSTOŚCI CECH

Podstawowym celem tych metod jest wywołanie wrażenia prawdziwie trójwymiarowego obrazu. Obraz obejmuje punkty rozłożone w trójwymiarowej przestrzeni i przedstawia je z intensywnością proporcjonalną do gęstości cech odpowiadających im treli. Wspólną cechą tych metod jest to, że zawierają w obrazie pełną informację o obiekcie występującą w plastrach; dają również wrażenie głębi i stereoskopii.

Do tego rodzaju systemów należą systemy z lustrami. Składają się one z czterech członów (rys. 3):

- urządzenia projekcyjnego,
- zestawu luster,
- ekranu,
- układu sterującego.

Zasada działania jest następująca. Plastry sceny są rzutowane na ekran w kolejności: **1, 2, ..., Z-1, Z, Z-1, Z-2, ...2, 1, ...**



Rys. 3. Systemy z lustrami

Zauważenie plastra na ekranie zależy od położenia ekranu i lustra. To obserwowalne położenie może być zmienione przez zmianę położenia ekranu lub rzutowanie na stały ekran i zastosowanie elastycznego lustra, które zmienia się z wklęsłego, przez płaskie, do wypukłego i z powrotem. Jeżeli położenie ruchomego elementu systemu jest zsynchronizowane z wyświetlaniem ciągu plastrów, to obserwator odbiera całościowy obraz, będący syntezą przestrzenną informacji obrazowej na wszystkich eksponowanych plastrach. Znane są również systemy z płaskimi wirującymi lustrami.

Systemy z lustrami mają dwie podstawowe wady:

- wymagają dużej szybkości przetwarzania danych oraz dużej częstotliwości ekspozycji plastrów, niezbędnej do „zlewania się" plastrów w trójwymiarowy obraz,
- występuje zaciemnienie środkowych obszarów sceny przez gęstości przednich i tylnych obiektów.

Znane są również metody oparte na technice holograficznej. Głównym problemem w tej technice jest przygotowanie odpowiednich holograficznych płatów, na których jest kodowana scena. Ponieważ jest to proces długotrwały, nie jest możliwe nie tylko w pełni automatyczne, ale nawet dialogowe manipulowanie scenami i obiektami.

KOMPUTEROWE PRZETWARZANIE SCENY I JEJ OBRAZOWANIE

Do obrazowania trójwymiarowej informacji o scenie stosuje się techniki reorganizacji sceny oraz techniki prezentowania powierzchni obiektu.

Techniki reorganizacji sceny

W technikach tych scena jest przekształcana w taki sposób, aby otrzymać odmienny od wyjściowego zbiór dwuwymiarowych informacji, które mogą być bardziej przydatne do określonych potrzeb. Techniki te można podzielić na dwie podgrupy: **techniki przeplastrowania sceny** (ang. reslicing) oraz **techniki przerzutowania sceny** (ang. reprojection).

Ideą **techniki przeplastrowywania sceny** jest wytworzenie zbioru plastrów z danej sceny, odpowiadających zbiorowi dowolnie skierowanych płaszczyzn tnących, bez przeprowadzania ponownego omiotu sceny. Poszczególne metody różnią się sposobem przeprowadzenia „cięcia" podzbioru treli, czyli dyskretnej trójwymiarowej sceny, przez ponownie zdefiniowane płaszczyzny tnące. Techniki tego rodzaju są przydatne wówczas, gdy omiatacz tomografu może wykonywać pomiar jedynie w pewnym położeniu.

Techniki przerzutowania sceny pozwalają wyeliminować efekt zaciemnienia, wynikający z nakładania się obrazów poszczególnych obiektów sceny. Gęstość w dowolnym punkcie płaszczyzny projekcyjnej jest obliczana metodą sumy ważonych gęstości treli w scenie, wzdłuż linii prostopadłej do płaszczyzny projekcyjnej w tym punkcie. Właściwy dobór wag w odpowiednich podobszarach sceny pozwala usunąć zupełnie efekt zaciemnienia, a przez to całkowicie odsłonić interesujące struktury sceny. Obrazy przerzutowane, obliczane dla kolejnych położeń płaszczyzny rzutowej, mogą być eksponowane kolejno po sobie z określoną szybkością, co daje wrażenie głębi.

Techniki obrazowania powierzchni ograniczających dyskretne obiekty

Umożliwiają one opisanie wyglądu zewnętrznego powierzchni obiektu. Są naturalnym rozwinięciem badań w kierunku komputerowego widzenia.

Techniki te zawierają następujące etapy przetwarzania obiektu:

- identyfikowanie,
- opisanie,
- sformowanie powierzchni,
- obrazowanie.

Automatyczne identyfikowanie obiektu jest zadaniem złożonym, dlatego większość z istniejących procedur obrazowania powierzchni identyfikuje obiekty sposobem konwersacyjnym lub półkonwersacyjnym. Natomiast złożoność algorytmów formowania i obrazowania powierzchni obiektu w dużym stopniu zależy od sposobu reprezentacji obiektu. Ponieważ informacja powierzchniowa jest trójwymiarowa, do jej obrazowania na płaskim ekranie jest wymagane jej przetworzenie. Ze względu na metody używane do opisu powierzchni wyróżnia się następujące techniki obrazowania obiektów za pośrednictwem obrazowania powierzchni ograniczających te obiekty:

- **schemat jednowymiarowy** (brzegowa reprezentacja powierzchni) – powierzchnię tworzy zbiór 1W brzegów na plastrach,
- **schemat dwuwymiarowy** (płatowa reprezentacja powierzchni) – powierzchnię tworzy zbiór płatów bądź innych elementów powierzchniowych,
- **schemat trójwymiarowy** – powierzchnię tworzy zbiór powierzchniowych trójwymiarowych geometrycznych elementów pierwotnych, tj. 3W – pierwotników geometrycznych.

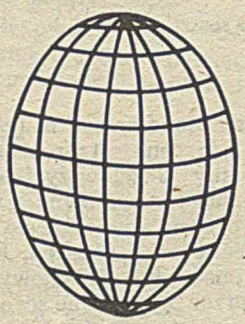
REPREZENTOWANIE POWIERZCHNI

Poniżej omówiono trzy metody reprezentowania powierzchni.

Brzegowa reprezentacja powierzchni

Schemat brzegowej reprezentacji powierzchni opiera się na obserwacji, że przecięcie powierzchni obiektu z **k**-tym plastrem sceny jest brzegiem tego obiektu na **k**-tym plastrze. Stąd powierzchnie obiektu są reprezentowane przez stos brzegów. Koncepcja ta jest znana z konwencjonalnej (generacyjnej) grafiki komputerowej.

Identyfikowania obiektu dokonuje się przez progowanie sceny. W ogólnym wypadku rozkład gęstości treli nie wystarcza do jednoznacznego zidentyfikowania obiektu. Pełne określenie obiektu musi bowiem zawierać rozkład gęstości cech w obiekcie, strukturalne zależności między podobszarami w obiekcie oraz topologiczne właściwości obiektu. Stos brzegów może być określony za pomocą algorytmu śledzenia brzegów, zastosowanego do każdego plastra sprogowanej sceny. Stos ten może być obracany w trójwymiarowej przestrzeni, w celu doprowadzenia obiektu do wymaganej orientacji. Natomiast zobrazowanie obiektu wymaga wykonania projekcji obróconego stosu na ekran. Otrzymuje się wówczas tzw. szkieletową reprezentację obiektu (rys. 4).



Rys. 4. Szkieletowa reprezentacja obiektu

Dla obiektów o złożonych kształtach szkieletowa reprezentacja powierzchni może być nieodpowiednia, gdyż nie zapewnia precyzji kształtów. Jest ona jednak prostsza obliczeniowo od dwóch pozostałych metod.

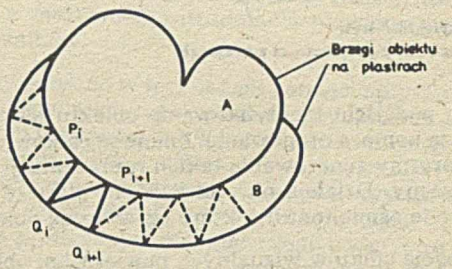
Płatowa reprezentacja powierzchni

Płatowa reprezentacja powierzchni jest techniką znaną z konwencjonalnej grafiki komputerowej i często w niej stosowaną. Może być również stosowana do obrazowania dyskretnej trójwymiarowej sceny. Powierzchnia jest przedstawiana w postaci połączonych elementów powierzchniowych, zwanych płatami: trójkątnych, czworokątnych lub bardziej złożonych.

Stosuje się dwa koncepcyjnie różne podejścia do prezentowania powierzchni według schematu dwuwymiarowego: metodę „plaster za plastrem" (ang. slice by slice) oraz metodę rzeczywiście trójwymiarowego obrazu.

W metodzie „plaster za plastrem" obiekt jest reprezentowany przez stos brzegów, jak w metodzie według schematu jednowymiarowego. Stos ten można otrzymać stosując algorytm śledzenia brzegu do dwuwartościowej sceny. Podobnie identyfikowanie obiektu może być dokonywane za pomocą programowania, musi być jednak zrealizowane całkowicie dialogowo i jest zwykle bardzo pracochłonne, zwłaszcza gdy scena zawiera wiele plastrów.

Następnym krokiem w tej metodzie jest formowanie powierzchni obiektu. Stosuje się w tym celu różne techniki, na przykład tzw. krycie dachówką między parami brzegów. Dość często stosuje się też tzw. trójkątowanie (ang. triangulation; rys. 5). Złożoność dachówkowania powierzchni zależy od przyjętego kryterium „akceptowalności powierzchni".



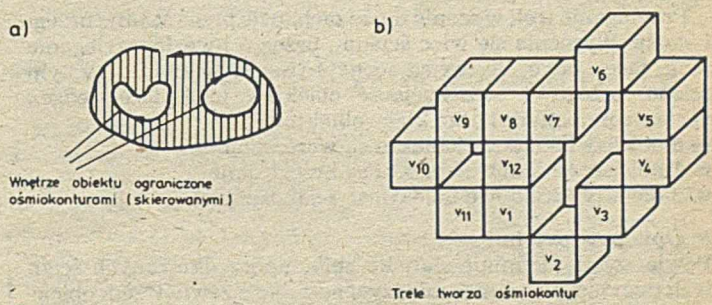
Rys. 5. Trójkątowanie powierzchni

Kolejnym etapem jest obrazowanie powierzchni. Aby powstał obraz stereo należy wydobyć ze sceny efekty wizualne, stosując następujące standardowe techniki:

- usunięcie zasłoniętych części powierzchni,
- cieniowanie widocznych części,
- rzut perspektywiczny obiektu,
- ruch kątowy obiektu,
- przenikanie się obiektów,
- teksturowanie powierzchni,
- efekty przezroczystości.

Jakość prezentacji obiektów za pomocą techniki „plaster za plastrem" zależy w dużym stopniu od algorytmów przetwarzania powierzchni i nie zawsze daje wizualnie akceptowalną powierzchnię. Zastosowanie techniki konwersacyjnej do rozwiązania powyższych problemów w złożonej scenie, może okazać się nie do przyjęcia ze względu na uciążliwość.

Metoda skierowanych konturów (ang. directed contour) polega na tworzeniu powierzchni obiektu ze ścian treli. Obiekt jest reprezentowany przez zbiór tzw. ośmiokonturów (ang. eight-contour), będących dyskretnymi analogami płaskich nie przecinających się krzywych. Ośmiokontury ograniczają obiekt i są powiązane jednoznacznie z wnętrzem obiektu kierunkiem (rys. 6).

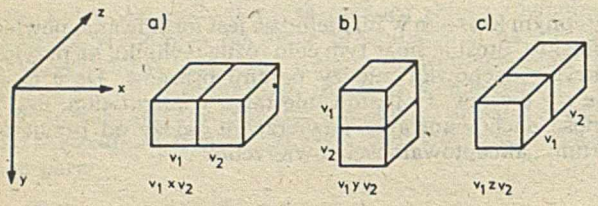


Rys. 6. Reprezentacja obiektów za pomocą skierowanych konturów

Do modelu sceny i obiektu wprowadza się pojęcia **x−**, **y−** i **z−przyległości** treli (ang. adjacency). Trele $v(x_1,y_1,z_1)$, $v(x_2, y_2, z_2)$, są (rys. 7):

- **x−przyległe,**
  gdy $|x_1 - x_2| = 1$ oraz $y_1 = y_2, z_1 = z_2$
- **y−przyległe,**
  gdy $|y_1 - y_2| = 1$ oraz $x_1 = x_2, z_1 = z_2$
- **z−przyległe,**
  gdy $|z_1 - z_2| = 1$ oraz $x_1 = x_2, y_1 = y_2$.

Intuicyjnie brzeg obiektu tworzą te trele, które należą do wnętrza obiektu oraz przylegają do zewnętrza obiektu przynajmniej jedną ścianą. Zbiór ścian obiektu tworzy się z odpowiednich ścian granicznych treli (tworzących ośmiokontur) na poszczególnych plastrach sceny.



Rys. 7. Przyległość treli
a) x-przyległe, b) y-przyległe, c) z-przyległe

W tym podejściu identyfikowanie obiektu jest dokonywane również za pomocą progowania. Znane są całkowicie automatyczne algorytmy generowania takich reprezentacji z dwuwartościowej sceny. Działają one na każdym plastrze wykrywając wszystkie te ośmiokontury, które ograniczają obiekt.

Wydobycie efektów wizualnych powierzchni obiektu jest podobne jak w technice „dachówkowania". Dla danego zbioru ośmiokontrów jest jeden zbiór tworzący ograniczającą obiekt powierzchnię i nie są niezbędne żadne dodatkowe badania tej powierzchni. Algorytm jest całkowicie równoległy, a jego złożoność jest funkcją jedynie liczby treli w zbiorze ośmiokonturów.

Metoda powierzchni trelowych (ang. cuberille) stanowi całkowicie odmienne od dotychczasowych podejście do obrazowania trójwymiarowej informacji o scenach w oparciu o obrazowanie powierzchni. Jest ono 3W w tym sensie, że dzieli splastrowaną scenę na obiekty jako całości. Do definiowania powierzchni obiektu stosuje się podział trójwymiarowej sceny. Identyfikowanie obiektu odbywa się konwersacyjnie za pomocą techniki progowania, jak podczas detekcji powierzchni. Natomiast metody reprezentacji obiektu, formowania powierzchni obiektu i wydobycie efektów wizualnych są tutaj całkowicie nowe.

Poszczególne etapy przetwarzania sceny są następujące.

- **Opis obiektów**

Podejście to daje precyzyjny opis obiektu i jego powierzchni, współgrający z intuicyjnym wyobrażeniem obiektów i powierzchni. Wykorzystuje się do tego specjalnie opracowaną topologię dyskretną. Wprowadza się pojęcie wnętrza sceny oraz wyróżnia dwie właściwości obiektów: **ograniczoność** i **spójność**. Obiekty są definiowane za pomocą progowania. Przed progowaniem każdy trel jest traktowany jako najmniejsza całość. Spójność jest definiowana na podstawie przylegania treli.

Przyleganie treli w scenie może zachodzić przez ściany, brzegi i węzły. Wyróżnia się więc ściano-, brzego- i węzłoprzyleganie (ang. face-, edge, vertex-adjacency) treli, a w związku z tym ściano, brzego- i węzłospójność obiektów (ang. face-, edge-, vertex-connectivity) oraz klasy obiektów:
- 0-obiekty (składowe obiektu są węzłospójne),
- 1-obiekty (składowe obiektu są brzegospójne),
- 2-obiekty (składowe obiektu są ścianospójne).

- **Opis powierzchni**

Powierzchnię definiuje się jako zbiór uporządkowanych ścian treli tworzących granicę między wnętrzem a zewnętrzem obiektu. Definicja ta jest zgodna z intuicyjnym wyobrażeniem powierzchni, gdyż:
- granica ma dobrze zdefiniowaną stronę wewnętrzną i zewnętrzną,
- granice są spójnymi składowymi zewnętrza obiektów,
- wykazano, że jeżeli wnętrze i zewnętrze obiektu są do siebie przyległe, to granica jest skończona i spójna oraz nie jest porowata.

- **Detekcja powierzchni**

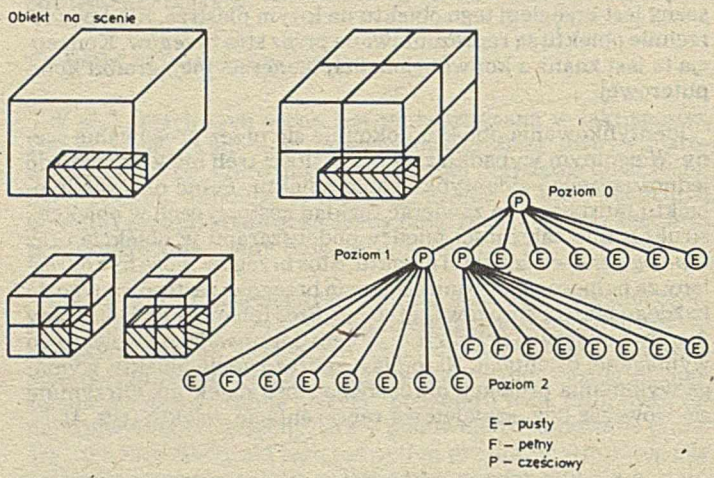
Problem detekcji powierzchni można sformułować używając pojęcia granicy. Obiekt jest zbiorem wszystkich tych treli, które mają gęstość **1** w sprogowanej dwuwartościwej scenie. Brzegospójny obiekt i jego granica są traktowane jako zbiór ścian między trelami należącymi do wnętrza i trelami należącymi do zewnętrza. Stosuje się również grafowy model do sformułowania i rozwiązania problemu detekcji granicy, w którym granica jest reprezentowana za pomocą tzw. granicznego binarnego drzewa.

- **Wydobywanie efektów wizualnych**

Do polepszenia efektów wizualnych wyświetlania powierzchni stosuje się takie same techniki jak w poprzednich metodach. Inne są jednak bardziej złożone algorytmy generowania tych efektów.

Reprezentacja obiektu za pomocą pierwotników graficznych

Metoda obrazowania obiektu za pomocą 3W pierwotników geometrycznych powstała przede wszystkim dla potrzeb projektowania wspomaganego komputerowo oraz komputerowego widzenia. W obrazowaniu trójwymiarowych dyskretnych scen szczególnie przydatna jest tzw. metoda równoległościanu (ang. parallel-piped). Opiera się ona na pewnej technice kodowania, tzw. 8-gałęzieniu dwuwartościowej sceny (ang.octree encoding).



Rys. 8. Operacja 8-gałęzienia sceny

Jej idea została przedstawiona na rys. 8. Jeżeli wszystkie trele w scenie mają wartość **1**, to scena jest reprezentowana za pomocą węzła oznaczonego „pełny". Jeżeli wszystkie one mają wartość **0**, to scena jest reprezentowana za pomocą węzła oznaczonego „pusty". W przeciwnym razie etykietuje się węzeł „częściowy" i kolejno dzieli się scenę na 8 podscen, które z kolei będą oznaczone: „pełny", „pusty" lub „częściowy". Identyfikowania obiektu dokonuje się przez progowanie sceny. Wynikowa dwuwartościowa scena jest wówczas reprezentowana przez ośmiorne drzewo. Obrazowanie obiektu wykonuje się za pomocą ograniczających obiekt powierzchni, lecz nie przeprowadza się operacji detekcji i formowania tej powierzchni, a jedynie wyświetla się podsceny reprezentowane przez węzły oznaczone „pełny". Nie stosuje się również wydobywania efektów wizualnych.

* * *

Omówione zagadnienia są w stadium początkowych badań. Dlatego prezentowane techniki są jeszcze bardzo niedoskonałe i niewiele z nich nadaje się do całkowicie automatycznej realiza-

dokończenie na s. 20

WITOLD ABRAMOWICZ
Poznań

Nowy sposób pamiętania w systemach informacyjnych

Systemy informacyjne – bazy danych, systemy wyszukiwawcze, a nawet systemy ekspertowe – są zdominowane przez tekst. Jest on nośnikiem informacji, służy także do reprezentowania danych umożliwiających organizowanie gromadzonej informacji. Stwierdzenia tego nie osłabia wprowadzenie nowych technik dialogu człowieka z komputerem, takich jak: okna i piktogramy. Systemy CAD, dysponujące z reguły grafiką o dużej rozdzielczości, umożliwiają kosztem dużego wysiłku obliczeniowego transformację (np. rotację) gromadzonych obrazów, lecz animacja przerasta ich możliwości.

Umożliwienie przetwarzania sekwencjami ruchomych obrazów może zrewolucjonizować zastosowania systemów informacyjnych. Przetwarzanie to obejmuje zapamiętywanie, organizowanie, wyszukiwanie i redagowanie (aktualizowanie) gromadzonej informacji. Trudno wyobrazić sobie choćby część zastosowań systemów informacyjnych umożliwiających np. zapamiętanie filmów w sposób tak zorganizowany, że zapewniają znalezienie pożądanych ich fragmentów według kryteriów wskazanych przez użytkownika.

Również przetwarzanie informacji dźwiękowej, np. w postaci głosu ludzkiego, nie jest praktycznie stosowane w systemach informacyjnych. Przykładem mógłby być system informacyjny gromadzący wszystkie wykłady prowadzone w danej szkole wyższej. Student mógłby wyszukiwać interesujące go fragmenty (pośród wszystkich zgromadzonych wykładów) według wskazanych przez siebie kryteriów.

Przedstawione przykłady systemów informacyjnych mogą wydawać się fantastyczne. Główną przeszkodą w ich realizacji jest brak podstaw teoretycznych, pozwalających na realizację tykich systemów w istniejących warunkach technologicznych. Można ją usunąć z jednej strony przez tworzenie nowych algorytmów, z drugiej przez postęp technologiczny – opracowywanie nowych architektur, rozwój pamięci, tworzenie urządzeń wejścia-wyjścia dla przetwarzania obrazu i głosu.

PAMIĘĆ OPTYCZNA

W systemach informacyjnych stosuje się głównie pamięci magnetyczne, mające wiele wad. Ich pojemność może być wystarczająca, jeżeli w systemie informacyjnym gromadzony jest tekst. Jednak do gromadzenia obrazów i głosu, przy obecnym stanie naszej wiedzy o metodach kodowania pojemności pamięci magnetycznych, jest zdecydowanie niewystarczająca. Pamiętanie dużej ilości informacji na nośnikach magnetycznych może okazać się kosztowne. Nośnik o pojemności 1 MB kosztuje od 5 do 200 dolarów (w zależności od technologii). Mimo niewątpliwego postępu pamięci te są ciągle wrażliwe na działanie pola magnetycznego, kurzu czy urazy mechaniczne.

Począwszy od odkrycia właściwości spójnego promieniowania elektromagnetycznego – lasera, przewidziano jego zastosowanie do gromadzenia informacji. W końcu lat sześćdziesiątych wszystkie poważne firmy komputerowe prowadziły badania w tej dziedzinie. Dla pamięci optycznych ROM (zwanych także po angielsku write once read only) można wskazać wiele sukcesów – nawet handlowych.

Pamięci optyczne ROM mogą znaleźć wiele zastosowań, czasem zaskakujących. Jeden z amerykańskich koncernów samochodowych w sieci zakładów serwisowych zatrudnia licznych mechaników – imigrantów nie umiejących czytać po angielsku. Jako środek pomocniczy zastosowano płytę wizyjną. Na monitorze znajdującym się obok miejsca pracy mechanik obserwuje sytuacje, które ma wykonać przy naprawianym samochodzie. Korzyści z takiego systemu są ewidentne: racjonalizacja procesu naprawczego (mechanik dokonuje naprawy poprawnie, stosując optymalne środki, w możliwie krótkim czasie), oszczędności wynikające z możliwości zatrudnienia niewielu kwalifikowanych pracowników.

Prawdziwy przełom w technologii pamięci optycznych wynika z przygotowania koncepcji pamięci wielokrotnie zapisywalnej. Nośnikiem informacji jest warstwa materiału ferromagnetycznego (np. stop terbu i żelaza lub terbu, żelaza i kobaltu) umieszczona na płycie i chroniona warstwą wysokoodpornego, magnetycznie obojętnego tworzywa. W celu zapisywania informacji wiązką promieniowania laserowego podgrzewa się punkt na warstwie ferromagnetyku do temperatury około 200°C. Pod jej wpływem w nagrzewanym punkcie cząstki zostają uporządkowane magnetycznie. Mimo obniżenia temperatury materiał pozostaje w tym punkcie namagnetyzowany. Każdy z punktów, wskazywalnych promieniem lasera może być namagnetyzowany z różną polaryzacją – można zatem w ten sposób zapamiętać 1 bit. Zmiana zawartości pamięci odbywa się przez nowe określenie polaryzacji magnetycznej wskazanych punktów. Eksperymenty z wielomilionowym uaktualnieniem zawartości pamięci wykazały dużą trwałość takiej pamięci.

Odczytywanie zapisanej informacji odbywa się na innej zasadzie. Słaby spolaryzowany promień laserowy jest przesyłany w kierunku pamięci. Efekt Faradaya powoduje zmianę płaszczyzny polaryzacji promienia, pod wpływem namagnesowania warstwy ferromagnetyku. Obserwacja zmian płaszczyzny polaryzacji promienia, kierowanego na wskazane punkty ferromagnetyku, umożliwia odczytanie informacji.

Nowe pamięci działające w myśl opisanej zasady mają pojawić się na rynku w 1988 r. jako dyski wymienne (stosowane podobnie jak popularne teraz dyski elastyczne). Firma 3M zapowiada dyski 5,25″ o pojemności 500 MB (ang. erasable laser optical disk). Natomiast firma Verbatim obiecuje 3,5″ dyski optyczne o pojemności 40 MB i niewiarygodnie niskiej cenie 50 dolarów za sztukę (stacja dysków ma być dostępna za 300 dolarów).

Cena, pojemność i niezawodność deklarowane przez producentów pozwalają mieć nadzieję, że pamięci optyczne staną się nowym nośnikiem – uniwersalnym sposobem zapamiętywania obrazów (także tekstów) i głosów (także muzyki).

KONSEKWENCJE DLA SYSTEMÓW INFORMACYJNYCH

Mimo wieloletnich badań nad nowymi pamięciami nie są znane odpowiednie metody i środki wykorzystania pamięci o tak szerokich możliwościach w systemach informacyjnych. Z technicznego punktu widzenia stało się możliwe ekonomiczne przetwarzanie informacji gromadzonej w różnej postaci (tekstu, grafiki i głosu) na jednym nośniku. Jak daleko możliwości nowych pamięci odbiegają od technik stosowanych w systemach informacyjnych, wskazuje artykuł prof. Appelratha. Zaprezentowano w nim prace na najwyższym światowym poziomie. Jednak do wyszukiwania informacji przedstawionej

graficznie wykorzystuje się jej opis w postaci tekstu. Dopiero on stanowi przedmiot zainteresowania systemu wyszukiwawczego – umożliwia zastosowanie sprawdzonych metod (w tym wypadku systemów wyszukiwawczych).

Problem metody opisu wiedzy ma kluczowe znaczenie dla opisywania informacji gromadzonej w systemie informacyjnym oraz reprezentacji zainteresowań użytkowników, a w konsekwencji dla strategii wnioskowania (systemy ekspertowe) lub wyszukiwania (bazy danych i systemy wyszukiwawcze). Systemy informacyjne umożliwiają duże zróżnicowanie opisu wiedzy: bazy danych (np. DDL), systemy wyszukiwawcze (np. tezaurus), systemy ekspertowe (np. ramy – ang. frames), pod warunkiem jednak, że informacja ma postać tekstu. Nie może być jednak inaczej, ponieważ dopiero na początku lat osiemdziesiątych zainteresowano się zintegrowanymi edytorami tekstu i grafiki. Użytkownik korzystający z nich jest najczęściej zmuszony do kompromisu: duże możliwości graficzne edytora i duże zapotrzebowanie na pamięć albo prymitywne właściwości graficzne i zapotrzebowanie na pamięć zbliżoną do edytorów tekstu. Zintegrowane edytory stanowią dopiero pierwszy etap tworzenia metod opisu wiedzy – jej prezentację. Autorowi nie są natomiast znane edytory umożliwiające równoczesne redagowanie trzech równoważnych nośników informacji: tekstu, głosu, grafiki.

Oczywistą konsekwencją braku rozwiązań dla obu kluczowych problemów – sposobu reprezentowania wiedzy oraz metod jej przedstawiania i modyfikowania, jest nieumiejętność:

- opisywania informacji gromadzonych w postaci tekstu, grafiki i głosu,
- przedstawiania potrzeb informacyjnych użytkowników systemów informacyjnych w tak samo bogaty sposób, tworzenia strategii wyszukiwania lub wnioskowania.

KIERUNKI BADAŃ

Badania mające na celu pokonanie tych problemów i w konsekwencji tworzenie systemów informacyjnych gromadzących informację przedstawioną jednocześnie w postaci tekstu, grafiki i głosu, zmierzają w dwóch kierunkach.

- Badania podstawowe w dziedzinie metod opisu wiedzy niezależnych od sposobu jej przedstawienia.

Sukces w tych badaniach nie pociągnie za sobą zapewne bezpośredniego rozwoju systemów informacyjnych; będzie wymagać opracowania algorytmów automatycznego opisywania (indeksowania) informacji gromadzonej w systemach informacyjnych oraz przygotowania nowych strategii wnioskowania lub wyszukiwania.

- Opisanie informacji za pomocą tekstu.

W systemie informacyjnym powstają więc trójki: informacja, adres informacji, opis informacji.
System informacyjny zarządza informacją w postaci tekstu, grafiki lub głosu, ale tylko opisy informacji są używane do formułowania potrzeb informacyjnych użytkowników, wnioskowania lub wyszukiwania. Efektywność takiego rozwiązania zależy od jakości algorytmów opisywania za pomocą tekstu informacji dostępnej jako grafika lub głos oraz od jakości samych tekstowych systemów informacyjnych.

* * *

Pamięci optyczne ze względu na dużą pojemność, możliwość pamiętania informacji w postaci tekstu, grafiki i głosu otwierają nowe perspektywy przed systemami informacyjnymi. Metodologia systemów informacyjnych jest na razie niewystarczająca dla przetwarzania informacji gromadzonej w tej formie.

Naczelna Organizacja Techniczna, Rada Wojewódzka Federacji SNT NOT w Łodzi, Komitet Naukowo-Techniczny ds. Informatyki, Politechnika Łódzka – Instytut Informatyki, Polskie Towarzystwo Cybernetyczne, Polskie Towarzystwo Elektrotechniki Teoretycznej i Stosowanej

organizują

w dniach 14–15 grudnia 1988 r.
w Domu Technika w Łodzi, Pl. Komuny Paryskiej 5a
konferencję naukowo-techniczną nt.

MIKROPROCESORY W ELEKTROTECHNICE

Celem konferencji jest omówienie problematyki zastosowania mikroprocesorów w elektrotechnice – w badaniach naukowych i praktyce, np. w napędzie elektrycznym, elektroenergetyce, telekomunikacji, miernictwie elektrycznym itp.

Konferencja winna posłużyć jako forum do wymiany doświadczeń w zakresie teoretycznych i praktycznych aspektów wdrażania mikroprocesorów do systemów elektrotechnicznych, z uwzględnieniem problematyki kształcenia kadr inżynieryjno-technicznych zajmujących się tym zagadnieniem.

Przewiduje się wygłoszenie 5. referatów w sesji plenarnej oraz 40. komunikatów w poszczególnych grupach tematycznych.

Koszt uczestnictwa w konferencji wynosi 9500 zł od osoby. Opłata ta obejmuje koszty organizacyjne, druk materiałów, dodatkowe posiłki i napoje oraz inne wydatki przewidziane przez komitet organizacyjny. Opłata nie obejmuje kosztu noclegów i wyżywienia, które uczestnicy ponoszą we własnym zakresie. 30 października br. upływa termin wnoszenia opłat na konto:
RW NOT Łódź, Pl. Komuny Paryskiej 5a
NBP III OM Łódź nr 47034-3027-132

Materiały konferencyjne obejmują pełne teksty referatów wygłaszanych na sesji plenarnej oraz komunikatów. Warunkiem ich uzyskania jest dokonanie wpłaty za uczestnictwo w konferencji.

Uczestnicy, którzy nadeślą karty zgłoszenia w terminie do 30 października br. mają zarezerwowane noclegi w hotelu „Światowit"
Adres organizatorów: Instytut Informatyki Politechnika Łódzka ul. Piotrkowska 220, 90-469 Łódź

ATMAN
ZAKŁAD PROJEKTOWO WDROŻENIOWY
Jednostka gospodarki uspołecznionej
04-082 WARSZAWA UL.KRYPSKA 39
TEL. 13-25-61 TLX.812530 agro pl

OFERUJE:
* mikrokomputery zgodne z IBM PC/XT/AT w dowolnych konfiguracjach;
* urządzenia peryferyjne (drukarki, digitizery, plotery, streamery i inne);
* lokalne sieci komputerowe (instalacja u użytkownika);
* materiały eksploatacyjne (dyskietki, kasety do drukarek i streamerów, pisaków do ploterów i inne).

PROJEKTUJE I INSTALUJE:
* systemy informatyczne wspomagania zarządzania i produkcji;
* dołączanie urządzeń nietypowych do IBM PC/XT/AT (urządzeń taśmy papierowej, elektr. maszyn do pisania, nietypowych streamerów, skanerów, przetworników A/C, C/A i innych);
* polskie litery.

ZAPEWNIA:
-bezpłatne doradztwo w zakresie doboru sprzętu i oprogramowania;
-bezpłatne oprogramowanie systemowe i narzędziowe;
-12-miesięczny serwis gwarancyjny;
-odpłatny serwis pogwarancyjny;
-dostawy w terminie 7-14 dni.

TEL.13-25-61 TLX.812530 AGRO PL
EO/869/88

JAN BIELECKI
Warszawa

Turbo C

Włączanie warunkowe

W bieżącym odcinku omówiono pozostałe dyrektywy procesora.

Dyrektywy włączania warunkowego

Zgodnie ze swoją nazwą dyrektywy włączania warunkowego umożliwiają generowanie tekstów identycznych z wybranymi fragmentami tekstu źródłowego. W ogólnym przypadku, strukturalna dyrektywa włączania warunkowego ma postać:

```
#if exp
    tekst
#elif exp1
    tekst1
#elif exp
    tekst2
  ...
#elif expn
    tekstn
#else
    tekst
#endif
```

Dyrektywa ta rozpoczyna się od frazy # if i kończy frazą # endif. Między tą parą fraz może występować dowolną liczbę razy fraza #elif i co najwyżej raz fraza # else. Wyrażenia *exp* i exp_j występujące we frazach # if i # elif są wyrażeniami określającymi sposób generowania tekstu. Wyrażenia te, poza tym, iż są wyrażeniami stałymi całkowitymi, a więc takimi, z których każde może być zastąpione liczbą, nie mogą zawierać operatora sizeof ani operatora konwersji. Mogą natomiast zawierać podwyrażenia o postaci:

defined *id*
albo
defined *(id)*

reprezentujące liczbę 1 albo 0, stoswnie do tego, czy w chwili ich napotkania makrodefinicja, której identyfikator pokrywa się z *id*, jest zdefiniowana albo nie zdefiniowana.

Zinterpretowanie dyrektyw włączania warunkowego rozpoczyna się od wyznaczenia wartości danej reprezentowanej przez wyrażenie *exp*. Jeśli wartość ta jest różna od 0, to całą dyrektywę zastępuje $tekst_p$. W przeciwnym razie wyznacza się kolejno dane reprezentowane przez wyrażenie exp_i. Jeśli pewne z tych wyrażeń, np. exp_j, reprezentuje daną o wartości różnej od 0, to całą dyrektywę zastępuje $tekst_j$. Jeśli okaże się, że żadne z wyrażeń nie reprezentuje danej o wartości różnej od 0, to całą dyrektywę zastępuje $tekst_f$. Nic nie stoi na przeszkodzie, aby każdy z wymienionych tekstów zawierał dyrektywy włączania warunkowego.

Dyrektywa włączania warunkowego może także zostać zapisana w postaci uproszczonej, nie zawierającej fraz # elif lub nie zawierającej frazy # else. W celu zapewnienia zgodności z podstawową wersją języka dopuszcza się także, aby fraza:

if defined *(id)*

została przedstawiona w postaci:

ifdef *id*

oraz aby fraza:

if !defined *(id)*

została przedstawiona w postaci:

ifndef *id*

Przykłady

A. Prosta dyrektywa włączania warunkowego:

```
# if defined(ArrFlg)
    int Arr[30];
# endif
```

Jeśli podczas interpretowania dyrektywy jest zdefiniowana makrodefinicja ArrFlg, to przytoczony napis wygeneruje napis:

```
int Arr[30];
```

W przeciwnym razie, nie zostanie wygenerowany żaden napis.

B. Dyrektywa włączania warunkowego z alternatywą:

```
#if a + b == 5
    printf ("%d", 5);
#else
    printf ("%d", 13);
#endif
```

Jeśli wyrażenie a + b = = 5 reprezentuje daną o wartości różnej od 0, to przytoczony napis wygeneruje napis:

```
printf ("%d", 5);
```

W przeciwnym razie przytoczony napis wygeneruje napis:

```
printf ("%d", 13)
```

C. Złożona dyrektywa włączania warunkowego:

```
#define Alfa 3
#define Beta(x) x - 4
#if Alfa == 6
    int One;
#elif Beta(2) + Alfa
    printf ("Izabela");
#endif
```

Zinterpretowanie przytoczonego napisu spowoduje wygenerowanie napisu:

```
printf ("Izabela");
```

D. Dyrektywy zagnieżdżone:

```
# define Alfa 5
# if Alfa * 5 > 20
    main()
#  if Alfa == 4
     int Arr[2];
#  elif Alfa == 3
     char Arr[2];
#  else
     {
#  endif
#  if 0
     printf ("Iza");
```

```
# else
    printf ("Kaja");
# endif
#else
    printf ("Ewa");
#endif
    }
```

Zinterpretowanie przytoczonego napisu spowoduje wygenerowanie napisu:

```
main()
{
  printf ("Kaja");
}
```

Dyrektywy pomocnicze

Dyrektywami pomocniczymi są: dyrektywa pusta oraz dyrektywy # line, # error i #pragma.

Dyrektywa # line musi być zapisana w taki sposób, aby po ewentualnym rozwinięciu zawartych w niej makrowywołań miała postać:

line *liczba literał-napisowy*

Zinterpretowanie dyrektywy # line dostarcza kompilatorowi informacji, że najbliższa instrukcja programu źródłowego pochodzi ze zbioru o nazwie określonej przez *literał-napisowy* i ma numer kolejny określony przez *liczbę*.

Dyrektywa # error ma postać:

error *napis*

Zinterpretowanie dyrektywy # error powoduje zaniechanie dalszego przetwarzania wstępnego wyprowadzenia komunikatu:

Fatal: *file line* Error directive: *napis*

w którym *file* jest nazwą zbioru poddawanego przetwarzaniu wstępnemu, a *line* jest numerem wiersza w tym zbiorze. *Na skutek błędu implementacji, preprocesor języka Turbo C zastępuje słowo* Fatal *słowem* Error.

Dyrektywa # pragma ma m. in. postać:

pragma inline

Zinterpretowanie takiej dyrektywy dostarcza kompilatorowi informacji, że program źródłowy zawiera wstawki asemblerowe.

W analogiczny sposób jest traktowana dyrektywa:

#pragma warn *napis*

Zawarty w niej *napis* może mieć postać *+www*, *−www* albo *.www*, w której *www* jest nazwą błędu sygnalizowanego przez kompilator. Użycie napisu *+www* powoduje włączenie ostrzegania, użycie napisu *−www* powoduje wyłączenie ostrzegania, a użycie napisu *.www* powoduje przyjęcie takiego sposobu ostrzegania, jaki obowiązywał w momencie podjęcia kompilowania programu.

Przykład

```
#line 44 "JanB.C"
#error That is all
```

Zinterpretowanie przytoczonego napisu powoduje zaniechanie dalszego przetwarzania wstępnego i wyprowadzenie komunikatu:

Fatal: JanB.C 44 Error directive: That is all

Poza makrodefinicjami, które można tworzyć za pomocą dyrektyw #define, preprocesor języka Turbo C zawiera szereg makrodefinicji predefiniowanych. Każda z nich rozpoczyna się i kończy parą znaków podkreślenia:

__LINE__ generuje liczbę określającą bieżący numer wiersza źródłowego;
__FILE__ generuje literał napisowy określający bieżącą nazwę zbioru źródłowego;
__DATE__ generuje literał napisowy o postaci "Mmm dd yyyy", określający bieżącą datę, np. "Dec 13 1981",
__TIME__ generuje literał napisowy o postaci "hh:mm:ss", określający bieżącą porę dnia;
__PASCAL__ generuje liczbę 1 albo 0, stosownie do tego, czy kompilacja odbywa się z domniemanym modyfikatorem pascal czy cdecl,
__CDECL__ generuje liczbę 1 albo 0, stosownie do tego, czy kompilacja odbywa się z domniemanym modyfikatorem cdecl czy pascal.

Poza wymienionymi makrodefinicjami jest w każdej kompilacji zdefiniowana dokładnie jedna z makrodefinicji: __TINY__, SMALL__, MEDIUM__, COMPACT__, LARGE__ I __HUGE__. Odwołanie się do niej umożliwia rozpoznanie aktualnie stosowanego modelu pamięci.

Przykład

```
#include <stdio.h>
#if defined (__PASCAL__)
cdecl
#endif
main()
{
    char *Model =
#if defined (__TINY__)
    "Tiny"
#elif defined (__MEDIUM__)
    "Medium"
#elif defined __COMPACT__)
    "Compact"
#elif defined (__LARGE__)
    "Large"
#elif defined __HUGE__)
    "Huge"
#endif
    ;
    printf (Model);
}
```

Program jest poprawny, niezależnie od tego, czy kompilacja odbędzie się z domniemanym modyfikatorem cdecl czy z modyfikatorem pascal. Jego wykonanie powoduje określenie przyjętego modelu pamięci.

Techniki obrazowania informacji o trójwymiarowych scenach

dokończenie ze s. 16

cji. Która metoda jest najlepsza – trudno dzisiaj powiedzieć. To zależy od zastosowania. Gdy istotniejsza jest informacja strukturalna i kształtowa, np. do wykrycia określonej anomalii, możliwość wyświetlania powierzchni wydaje się być bardziej użyteczna. Praktycznych wyników stosowania tych metod jest, jak dotychczas, niewiele. Nie w pełni znane są ich pozytywne i negatywne cechy.

Która metoda gwarantuje lepszy obraz – również trudno określić. Zależy to od stawianych wymagań. Metody te są ciągle doskonalone i rozwijane. Zagadnienie percepcji informacji zawartej w trójwymiarowej scenie jest na tyle ważne, że bez jego rozwiązania trudno sobie wyobrazić dalszy rozwój współczesnej robotyki, informatyki i innych pokrewnych dziedzin wiedzy.

LITERATURA

[1] Artzy E., Frieder G., Herman G. T.: The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm. Computer Graphics and Image Processing, Vol. 15, pp. 1–24, 1981
[2] Liu H. K.: Two- and three-dimensional boundary detection. Computer Graphics and Image Processing, Vol. 6, pp. 123-134, 1977
[3] Udupa J. K.: Interactive segmentation and boundary surface formation for 3D digital images. Computer Graphics and Image Processing, Vol. 18, pp. 213-235, 1982
[4] Udupa J. K.: Display of 3D information in discrete 3D scenes produced by computerized tomography. Proc. of the IEEE, Vol. 71, No. 3, pp. 420–432, 1983.

ANDRZEJ TUROWICZ
Ośrodek Informatyki
Urząd Wojewódzki
Wrocław

Lokalna sieć komputerowa D-Link (2)

Po ogólnych informacjach przedstawionych w pierwszej części artykułu omówię teraz przykład praktycznej sieci.

PRZYKŁAD

Załóżmy, że w pewnym przedsiębiorstwie skomputeryzowano ewidencję wyrobów gotowych w magazynie oraz ewidencję ich sprzedaży. W tym celu zainstalowano mikrokomputery w dziale sprzedaży, w księgowości i w magazynie. Każdy z działów wykonuje inny rodzaj przetwarzania danych na tych samych zbiorach. Projektant systemu postanowił rozmieścić zbiory danych oraz komputery zgodnie z tabelą 1, w której **FD** oznacza stację dysków elastycznych 360 KB, a **HD10** i **HD20** – dysk stały o pojemności odpowiednio 10 i 20 MB.

Tabela 1. Rozmieszczenie komputerów i ich funkcje

Nr komputera	Dział przedsiębiostwa	Zbiór informacji	Pamięci dyskowe
1	Dział sprzedaży	Odbiorcy wyrobów (2MB)	HD10, FD
2	Dział księgowości	Katalog wyrobów (5MB) obroty miesięczne (8MB)	HD20, FD
3	Magazyn wyrobów	Katalog stanów (6MB) oprogramowanie (360KB)	HD20 FD
4	Magazyn wyrobów	Katalog stanów (360KB)	FD, FD

Oprogramowanie systemu dla komputerów w magazynie znajduje się na dysku komputera numer 3 i ma do niego dostęp komputer numer 4. Pozostałe programy znajdują się w tych komputerach, które je wykonują. Baza danych dBase III Plus jest umieszczona na dysku komputera w księgowości. Komputer w księgowości oraz komputer w dziale sprzedaży mają własne drukarki. Komputery z magazynu mają korzystać z drukarek w księgowości.

Komputer w dziale sprzedaży ma zadanie wprowadzać informację o sprzedawanych wyrobach z dokumentu **Wz.** Sprawdza, czy istnieje wymagana liczba wyrobów w magazynie, tj. w pliku Katalog Stanów komputera numer 3, oraz czy istnieje odbiorca w pliku Katalog Odbiorców komputera numer 1. Następnie zapisuje transakcję w pliku Obroty komputera numer 2, poprawia stan w Katalogu Stanów komputera numer 3 i drukuje fakturę. Dane takie, jak nazwa i cena wyrobu, pobiera z pliku Katalog komputera numer 2.

Komputer numer 2 w księgowości ma zadanie aktualizować plik Katalog Wyrobów, a także przechowuje plik Obroty Miesięczne i wykonuje okresowe zestawienia.

Komputer numer 3 i komputer numer 4 rejestrują przychody i rozchody w magazynie wyrobów gotowych. Wszystkie operacje dotyczące stanów są dokonywane na dysku komputera numer 3. Komputer numer 4 sam nie przechowuje plików, jest więc terminalem sieci komputerowej. Magazyn jest obsługiwany przez dwa stanowiska komputerowe. Wynika to z natężenia transakcji wykonywanych przez obsługę. Projektant systemu rozmieścił pliki na dyskach stałych komputerów sieci zgodnie z tabelą 2.

Tabela 2. Rozmieszczenie plików na dyskach stałych

Nr komputera	Plik	Dysk skorowidz	Nazwa pliku
1	Odbiorcy wyrobów	c:/ow	Odbiorca.dbf
2	Katalog wyrobów,	c:/kw	wyroby.dbf
	obroty miesięczne,	c:/om	obroty.dbf
	dBase III Plus	c:/dbase	...
3	Katalog stanów,	c:/ks	stan.dbf
	oprogramowanie magazynu	v1	

Oprogramowanie magazynu dla komputerów 3 i 4 znajduje się na woluminie numer 1 (v1), na dysku **c** komputera numer 3. Każdy z komputerów powinien posługiwać się tylko tą częścią dysku innego komputera, która jest niezbędne do wykonywania jego zadań. Dlatego przy częściowym przetwarzaniu skorowidza dysku komputery powinny skorzystać z polecenia SUBST systemu DOS.

BUFOR DRUKARKI NA DYSKU STAŁYM

Każdy komputer może mieć swoją własną drukarkę, jednak w dużej sieci jest to rozwiązanie kosztowne. Z drugiej strony nie można sobie wyobrazić pracy bez papierowych zestawień lub kopiowania zawartości ekranów. Urzędnik, naukowiec lub inżynier musi mieć możliwość wydrukowania wyników swoich obliczeń.

W sieci mogą pracować komputery z różnymi drukarkami, różniącymi się szybkością pracy i jakością druku. Mogą też być komputery bez drukarek – sieć umożliwia kierowanie wydruków do tych drukarek, które są dla danego wydruku odpowiednie. Pojawia się więc problem kolizji komputerów pragnących drukować na tej samej drukarce.

Problem ten rozwiązano w sposób następujący. Komputer, który pozwala drukować sieci na swojej drukarce, rezerwuje na dysku obszar buforowy na wydruki. Wydruki własne oraz pochodzące od innych komputerów z sieci są automatycznie kierowane do bufora i wyprowadzane w kolejności zgłoszeń. Komputer-właściciel może badać statystykę pracy drukarki, kolejkować i anulować wydruki. Nazywa się to systemem gromadzenia wydruków (ang. print spooler). Wydruki są rozdzielone stroną tytułową, zawierającą informacje o nazwie zadania, numerze komputera i numerze grupy komputerów, datę i czas drukowania. Aby udostępnić tę usługę na danym komputerze należy wykonać polecenie **SPL**. Program pyta o nazwę bufora drukarki na dysku; można podać własną nazwę lub nazwę **SPL.BUF**, wymyśloną przez projektantów sieci D-Link. Następnie pyta się o nazwę strony tytułowej i o to, czy jest ona konieczna. Można utworzyć własną, o dowolnej nazwie, lub zastosować fabryczną o nazwie **SPL.SEP**. Następne pytania dotyczą wielkości bufora w kilobajtach, liczby użytkowników drukarki i wielkości kolejki wydruków. Po podaniu tych informacji program pyta, czy aktywować gromadzenie oraz czy zapisać informacje do pliku **AUTOEXEC.BAT**, w celu automatycznego inicjowania gromadzenia wydruków. Stronę tytułową można wykonać korzystając z dowolnego edytora tekstów, według zasad zawartych w dokumentacji.

Polecenie **SPLSTAT** informuje o kolejce wydruków. Wydruki pilne można przestawić na początek kolejki. Zastosowanie opcji **SPOOL** jest bardzo użyteczne i znacznie przyspiesza prace

programów nie tylko w sieci. Jest pewna mała niedogodność związana z czasem aktywacji bufora drukarki. Czas ten jest proporcjonalny do wielkości bufora i wynosi od kilkunastu do kilkudziesięciu sekund.

URZĄDZENIA ZDALNE

Komputer, musi jawnie pozwolić na pracę innych komputerów z sieci ze swoimi urządzeniami. Wyjątek stanowią woluminy, jeśli komputer je posiada. Praca z innymi urządzeniami zdalnego komputera wymaga:

- dołączenia się komputera zdalnego do sieci,
- dołączenia się komputera lokalnego do sieci,
- udostępnienia własnych urządzeń przez komputer zdalny,
- dołączenia urządzeń przez komputer lokalny.

Prawa dostępu nadaje się poleceniem **SETDEV**, a do dołączania urządzeń służy polecenie **CONNECT**. Każdy z komputerów sieci musi wydać zezwolenie na dostęp do swoich urządzeń przez inne komputery. W naszej sieci komputery będą korzystać z różnych urządzeń różnych komputerów, będą miały różne wymagania. Jedne będą tylko odczytywać i zapisywać na danym urządzeniu. Dla naszego przykładu sieci, cztery komputery powinny wydać odpowiednie polecenia, tak jak to podano na wydrukach 1–4.

Komputer nr 1	Komentarz
LOGON Sprzedaz	Rozpoczęcie zadania "Sprzedaz". Nazwa musi być jednoznaczna.
SETDEV H1 /G7/S255/Xw/Yw/Zr	Zezwolenie na użycie dysku H1. Właściciel: grupa 7, numer 255. Właściciel i grupa mają prawo do zapisu (Xw, Yw); inni tylko prawo odczytu (Zr).
CONNECT D/2/H1	Przydział dysku stałego nr 1 komputera 2 jako dysku D, tylko do odczytu.
CONNECT E/3/H1/W	Przydział dysku 1 komputera 3 jako dysku E, do odczytu i zapisu.
SUBST F: D:\dbase	Polecenie DOS; symulacja dysku F katalogiem D:\dbase, na dołączonym dysku komputera nr 2. Komputer pracuje z dyskiem F reprezentowanym jako fragment dysku C komputera 2
SUBST G: D:\kw	Jak wyżej, symulacja dysku G.
SUBST H: D:\om	Symulacja dysku H.
SUBST I: E:\ks	Symulacja dysku I.

Wydruk 1

Komputer nr 2	Komentarz
LOGON Ksiegowy	Rozpoczęcie zadania.
SETDEV H1 /G7/S255/Xw/Yw	Prawo dostępu do dysku H1. Właściciel: grupa 7. Prawo właściciela i grupy do zapisu.
SETDEV P1 /G7/S255/Xw/Yw	Zezwolenie na dostęp do drukarki, z prawem zapisu dla grupy 7.
CONNECT D/1/H1	Przydział dysku C komputera nr 1 do odczytu.
CONNECT E/3/H1	Przydział dysku C komputera nr 3.
SUBST F: D:\ow	Symulacja dysku F katalogiem ow dysku C komputera nr 1.
SUBST G: E:\ks	Symulacja dysku G katalogiem stanów z dysku C komputera nr 3.

Wydruk 2

Komputer nr 3	Komentarz
LOGON mag-1	Rozpoczęcie zadania.
SETDEV H1 /G7/S255/Xw/Yw	Zezwolenie na zapis na dysku C dla grupy 7.
CONNECT D/2/H1	Przydział dysku C komputera nr 2.
CONNECT LPT1/2/P1/W	Przydział drukarki P1 komputera nr 2 jako LPT1.
SUBST E: D:\dbase	Symulacja dysków E i F katalogami
SUBST F: D:\ks	z komputera nr 3.

Wydruk 3

Komputer nr 4	Komentarz
LOGON mag-2	Rozpoczęcie zadania.
CONNECT C/3/H1/W	Przydział dysku C komputera nr 3 do zapisu.
CONNECT D/2/H1	Przydział dysku C komputera nr 2.
CONNECT LPT1/2/P1/W	Przydział drukarki P1 komputera nr 2 jako LPT1.
SUBST E: D:\dbase	Symulacja dysków E i F katalogami
SUBST F: D:\ks	z komputera nr 3.
CONNECT G/3/v1	Przydział woluminu v1 z komputera 3 jako dysku G.

Wydruk 4

Należy zauważyć, że komputer numer 1 w dziale sprzedaży ma tylko jeden dysk stały **H1** – dysk **C**. Dyski **E, F, G, H** i **I** są symulowane przez system operacyjny DOS. Ponieważ DOS obsługuje tylko dyski od **A** do **G**, więc w pliku **CONFIG.SYS** należy umieścić polecenie **LASTDRIVE = I. O** – Oprogramowanie na woluminie **V1** nie wymaga zadeklarowania.

Wszystkie te polecenia można wydać z klawiatury, najlepiej jednak umieścić je w plikach **AUTOEXEC.BAT**. Automatyzacja procesów przydzielania pozwala uniknąć wielu błędów oraz zapewnia właściwą ochronę informacji.

W naszym przykładzie nie były stosowane hasła. Aby recepta była pełna, warto rozszerzyć projekt o drukarkę laserową, zainstalowaną w dziale sprzedaży. Eksploatacja tej drukarki jest dość kosztowna, kierownictwo przedsiębiorstwa podjęło więc decyzję o udostępnieniu jej tylko dla niektórych prac. Aby uzyskać dostęp do drukarki laserowej ustalono system haseł. Komputer w dziale sprzedaży wydaje polecenie:

SETDEV P2/AW/P

Oznacza to, że zezwala na użycie laserowej drukarki **P2** wszystkim tym komputerom, które podadzą właściwe hasło. O treść hasła komputer pyta obsługę w trakcje wykonywania polecenia. Jeśli, na przykład, operator napisał **X12cJk**, to od tego czasu hasło to daje prawo drukowania na drukarce **P2**. Każdy komputer, który będzie chciał pracować z tą drukarką musi wydać polecenie:

CONNECT LPT3/1/P2/W/P

przydzielające jako **LPT3** (można podać inny numer) drukarkę nr 2, z komputera nr 1, na zapis po podaniu właściwego hasła. Próba przydzielenia urządzenia bez hasła, to jest bez parametru /P, lub gdy hasło jest niewłaściwe kończy się odpowiedzią komputera:

Access denied Waiting...(or press (ESC) to abort)

co oznacza, że dostęp jest niemożliwy ze względu na brak prawa dostępu, brak hasła lub że hasło jest niewłaściwe.

* * *

Informacje przedstawione w dwóch pierwszych częściach artykułu wyczerpują w zasadzie wszystkie tajniki budowy sieci oraz zasad współpracy komputerów między sobą. W części trzeciej zostaną przedstawione instrukcje usługowe sieci D--Link.

Jak widać, praca w sieci nie jest współpracą mikroprocesorów i zadania są wykonywane w każdym komputerze osobno; można jedynie korzystać z urządzeń innych komputerów. W sieci można jeszcze stosować modemy, jako urządzenia dołączalne. Modem może łączyć komputery jednej sieci z drugą siecią lub z komputerem typu RIAD lub Odra. W instytucji autora sieć komputerów typu IBM PC jest połączona z komputerem Odra 1305. Odra pracuje w systemie George 3, a mikrokomputery w sieci D-Link z systemem operacyjnym DOS 3.10. Między komputerem Odra i mikrokomputerami są przesyłane pliki typu **TEXT** w znaczeniu IBM, a w znaczeniu ODRY typu GRAPHIC. Autorem programu komunikacji jest p. Marek Zgadzaj z NAVA--TRANS we Wrocławiu. Program umożliwia transmisję za pomocą modemów produkcji polskiej i innych. Transmisja jest asynchroniczna z szybkością od 300 do 9600 bitów na sekundę. Szybkość transmisji zależy od wyposażenia komputera Odra.

Uprzejmie informujemy Czytelników, że egzemplarze INFORMATYKI – bieżące i archiwalne – można kupić nie tylko w kioskach Ruchu, Klubie NOT SIGMY, Zakładzie Kolportażu i Dziale Handlowym (szczegóły podano w WARUNKACH PRENUMERATY), ale również
w lokalu naszej redakcji
ul. Mickiewicza 18 m. 17 w Warszawie, tel. 39-14-34
oraz
w specjalistycznej księgarni PP „Domu Książki"
ul. Mokotowska 51/53 w Warszawie, tel. 28-16-14
Zapraszamy wszystkich zainteresowanych.

Skrośny system firmy Alsys wspomagania programowania w Adzie dla PC/AT

System o nazwie iAPX86 Ada Cross-Development System firmy Alsys wspomaga tworzenie oprogramowania użytkowego na systemy wbudowane i czasu rzeczywistego, dla mikroprocesorów rodziny iAPX86 firmy Intel. Kompilator został atestowany do pracy zarówno w trybie wrodzonym, jak i skrośnym. Ma cechy systemu wykonawczego nie wymagającego obecności systemu operacyjnego. Zawiera wyrafinowane środowisko zarządzania bibliotekami, wspierające proces wytwarzania oprogramowania. Szerokie możliwości konfiguracyjne i programy narzędziowe do umieszczania segmentów programu w pamięci PROM, to następne cechy tego produktu. Pakiet zawiera również Ada-Probe – popularny program uruchomieniowy poziomu źródłowego, zaadoptowany do uruchamiania skrośnego na konfiguracji docelowej.

W wypadku mikroprocesorów 80286 i 80386 programy aplikacyjne mogą wykorzystywać tryb ochronny. Dzięki temu można w pełni używać 16-megabajtowej przestrzeni adresowej mikroprocesora 80286.

ŚRODOWISKA WYKONAWCZE

Programy składające się na system wykonawczy RTE (ang. run-time executive) pozwalają korzystać z wielozadaniowości, obsługi wyjątków, zarządzania pamięcią i operacji we-wy na komputerze jednopłytowym lub w środowisku DOS. Istnieje również jądro wykonawcze (ang. run-time kernel) współpracujące z systemem wykonawczym i udostępniające sprzężenie ze sprzętem w środowisku komputerów jednopłytowych. W środowisku systemu operacyjnego, sprzężenie ze sprzętem jest realizowane za pomocą procedur DOS-u.

Ada Cross-Development System udostępnia kilka środowisk testowych, stanowiących wygodny i elastyczny środek wspomagający wytwarzanie oprogramowania. Dzięki temu można:

- Kompilować program aplikacyjny na mikrokomputerze PC/AT i wykonać wstępne testy pod kontrolą PC-DOS (na PC/AT, PC/XT lub PC/1, w zależności od wybranego kodu).
- Testować programy aplikacyjne na PC/AT, wykorzystując kartę 80286 jako komputer jednopłytkowy. Tryb ten jest dostępny po skonsolidowaniu programu z jądrem wykonawczym. Program jest wykonywany na PC/AT bez udziału systemu operacyjnego, pod kontrolą jądra wykonawczego w taki sposób, w jaki byłby wykonywany na sprzęcie docelowym. Tym samym korzysta się ze środowiska jednopłytkowego bez ładowania skrośnego (ang. downloading). Firma Alsys udostępnia również narzędzia programowe (procedury i programy obsługi) niezbędne przy wykorzystywaniu karty procesora 80286 mikrokomputera PC/AT jako komputera jednopłytkowego.
- Załadować program do konfiguracji jednopłytkowej i wykonać go pod kontrolą systemu i jądra wykonawczego na sprzęcie docelowym.

System zawiera również programy usługowe komunikacji między komputerem macierzystym i docelowym.

WŁAŚCIWOŚCI SYSTEMU

W skład systemu wchodzą:

- dwa kompilatory – kompilator skrośny rodziny mikroprocesorów iAPX86 (8088, 8086, 80186, 80286, 80386) i kompilator wrodzony systemu operacyjnego,
- dwa programy uruchomieniowe – dwie wersje AdaProbe, do uruchomiania aplikacji[1] wbudowanych i do uruchomiania aplikacji DOS-u,
- narzędzia, do których należą biblioteki, zarządca bibliotek, zarządca jednostek, program wiążący (ang. binder), program formatujący (AdaReformat), analizator odwołań (AdaXref), konfigurator, usługowe programy ładowania skrośnego i programowania pamięci PROM,
- system wykonawczy i jądro wykonawcze, przeznaczone do sterowania wykonaniem w systemach jednopłytkowych i mikrokomputerowych typu PC,
- pełna dokumentacja ułatwiająca programowanie, kompilowanie, testowanie i uruchamianie.

Całość systemu napisano w Adzie. Kod kompilatora, programów narzędziowych, pierwotnie zdefiniowanych bibliotek i programów uruchomieniowych liczy ponad 500 000 wierszy.

System wykonawczy

Rozmiary docelowego systemu wykonawczego zmieniają się w zależności od aplikacji, ponieważ nie dołącza się niewykorzystanych podprogramów. Centralnym zagadnieniem projektowym systemu wykonawczego była jednak obsługa wyjątków. Ponieważ zwykłe sekwencje **CALL/RETURN** (dla procedury) i **ENTRY/EXIT** (dla bloku) nie powodują żadnego narzutu przy zapamiętywaniu informacji o stanie wyjątku, nie ma żadnych narzutów związanych z obsługą wyjątku dopóty, dopóki nie zostanie zgłoszony wyjątek.

System wykonawczy dokonuje efektywnego przydziału pamięci. Ułatwia dostęp, minimalizuje zajętość pamięci i bardzo sprawnie przeprowadza odzyskiwanie zwolnionej przestrzeni, natomiast kompilator tylko optymalizuje przydział i zwalnianie pamięci obiektom dynamicznym i tymczasowym. Odzyskiwanie pamięci zachodzi tak, jakby postać źródłowa zawierała pragmę **CONTROLLED**, dzięki czemu czas odzyskiwania jest określony bez narzutów i niepewności związanej z pełnym odzyskiwaniem pamięci w czasie wykonania.

System wykonawczy udostępnia także w całości mechanizmy wielozadaniowości Ady. Szeregowanie jest oparte na schemacie z wywłaszczaniem (ang. pre-emptive), a podział czasu podlega kontroli użytkownika. Na podstawie testów określono, że spotkanie przy użyciu parametrów na mikroprocesorze 80286, o częstotliwości zegara 8 MHz, zajmuje tylko 433 mikrosekundy, a synchronizowanie – 262 mikrosekundy.

Programy uruchomieniowe

AdaProbe składa się z symbolicznego programu uruchomieniowego poziomu źródłowego i tzw. wziernika (ang. program viewer). Może być użyty zarówno w trybie skrośnym do programów wykonywanych na komputerze jednopłytowym, jak i w trybie wrodzonym do programów wykonywanych na PC/AT pod kontrolą PC-DOS. W trybie skrośnym AdaProbe znajdując się na komputerze macierzystym PC/AT komunikuje się ze środowiskiem docelowym przez łącze szeregowe. W obydwu trybach, AdaProbe stanowi uniwersalne narzędzie znacznie przekraczające możliwościami tradycyjne programy uruchomieniowe.

Sprzężenie użytkowe dla AdaProbe jest w pełni symboliczne, na poziomie źródłowym. Obiekty lub instrukcje poddawane sprawdzeniu i operacje wglądania (ang. viewing) wybiera się wskazując odpowiednie miejsce programu źródłowego. W programie AdaProbe połączono klasyczne funkcje sterowania wykonaniem z nowymi mechanizmami wglądania. Funkcje sterujące wykonaniem obejmują punkty wstrzymania, pracę krokową, sprawdzenie i zmienianie wartości zmiennych.

Mechanizmy wglądania, ściśle powiązane z biblioteką programów Ady (ang. Ada program library), umożliwiają uzyskanie pełnych informacji o dowolnym obiekcie i wygenerowanie listy wszystkich miejsc programu,

[1] Nazwą aplikacja określa się program użytkowy (przyp. red.).

w których występują odwołania do obiektu. Można również sprawdzać miejsca deklaracji obiektów i dynamicznie zmieniać ilość podawanych informacji. AdaProbe jest bardzo pożytecznym narzędziem do analizowania programów aplikacyjnych nie tylko w trybie uruchomieniowym.

Inne narzędzia

Omawiany system można z łatwością konfigurować do wykonania na dowolnym procesorze rodziny iAPX86. Opis specyficznej płyty stanowiącej środowisko wykonawcze jest umieszczany na plikach modyfikowalnych przez użytkownika. Odpowiednie segmenty kodu w tych plikach są napisane w języku asemblera mikroprocesora 8086 i mogą być konsolidowane z jądrem wykonawczym przy użyciu programów usługowych firmy Intel. Istnieją przykładowe postacie plików dla iSBC 286/12 firmy Intel.

Oprócz wykonywanego przez system kompilujący eliminowania niewykorzystanych procedur czasu wykonania, przewidziano możliwość konsolidowania pisanych przez użytkownika procedur we-wy i obsługi urządzeń. Przy użyciu programów usługowych można określić rozmieszczenie segmentów w pamięci, określić ich typ (R/O lub R/W), sterować wielkością i położeniem segmentów stosu i stogu.

Kod wytworzony przez system można umieścić w pamięci PROM pojedynczej płyty, do wykonania bezpośrednio po wyzerowaniu pocesora.

Kompilowanie i wiązanie programów przeprowadza się przy użyciu poleceń o składni przypominającej Adę. System wyposażono w łatwo dostępną, wyczerpującą diagnostykę i interakcyjny samouczek. Wyczerpujące komunikaty o błędach, umieszczane na wydruku postaci źródłowej programu, wskazują miejsce wystąpienia błędu, lecz także zawierają dodatkowe sugestie co do jego usunięcia. Pełna diagnostyka polega na cytowaniu odpowiednich ustępów programu, co pozwala programiście znaleźć przyczyny wystąpienia błędu. Dokładniejsze wyjaśnienia diagnostyczne odsyłają programistę do odpowiednich części podręcznika Ady.

DOKUMENTACJA

Wraz z oprogramowaniem firma Alsys dostarcza kompletną dokumentację, w której przedstawiono zasady pracy i te aspekty języka, które umożliwiają efektywne wykorzystanie systemu.

W instrukcji instalowania opisano ładowanie kompilatora i ustawianie jego parametrów. Wyjaśniono zasady dołączania dodatkowej karty pamięci do mikrokomputera PC/AT.

Podręcznik użytkownika zawiera omówienie procesu kompilacji – wyjaśnia składnię poleceń kompilowania w środowisku systemu operacyjnego, wiązanie jednostek kompilacji, konsolidowanie powiązanych programów wynikowych z systemem wykonawczym **RTE**.

Podręcznik projektanta opisuje środowisko wielobiblioteczne. Zawiera również omówienie składni poleceń programów narzędziowych, zarządzających bibliotekami i ich zawartością.

Podręcznik programisty aplikacyjnego wyjaśnia takie rzeczy, jak sprzężenie z procedurami systemu operacyjnego, procedury obsługi wejść przerwań i wykorzystanie pragmy
INTERFACE (sprzężenie z językiem asemblera).

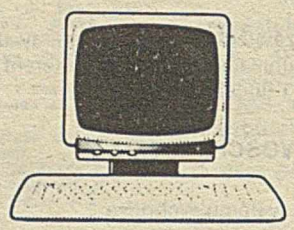
Podręcznik systemu skrośnego zawiera omówienie użycia kompilatora w trybie skrośnym. Można w nim znaleźć także szczegóły dotyczące konfigurowania i dopasowywania systemu wykonawczego.

Podręcznik Ady (ANSI/MIL-SID 1815A) zawiera definicje języka, a dodatek F podręcznika – informacje o właściwościach zależnych od implementacji, jak również opisy sprzężeń z innymi językami.

W dokumentacji umieszczono również zbiór programów Ady o nazwie **Ada Sampler** w celu ułatwienia nauki języka i zrozumienia jego ducha, a także – przykłady plików konfiguracyjnych dla procesora iSBC 286/12 firmy Intel.

Oprac.:
M. KUC

Dla użytkowników systemów ODRA i ICL



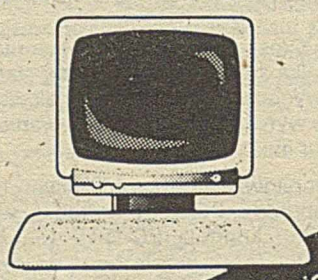
ZEKOM proponuje:

- TERMINAL EKRANOWY MV 2581
  przeznaczony do pracy w systemach ODRA 1300 wyposażonych w grupową jednostkę sterującą JSG-7802 jako odpowiednik monitora MERA-7911N
- TERMINAL EKRANOWY MV 2582E
  przeznaczony do pracy w systemach ODRA 1300 ICL 1900, ICL 2900, ICL System 4 jako odpowiednik monitora 7181 2 firmy ICL
- MULTIPLEKSER TX 82
  przeznaczony do wspolpracy z terminalami MV 2582E jako 8-kanalowy odpowiednik adaptera QLSA firmy ICL
- ADAPTER LINI TA 42
  przeznaczony do wspolpracy z terminalami MV 2582E jako 4-kanalowy odpowiednik adaptera LSA firmy ICL

ZEKOM
ZAKŁAD ELEKTRONIKI KOMPUTEROWEJ
Skr. pocztowa 35, 90-955 Łódź 8, tel. 57-25-83

E0/1088/88

ZAKŁAD ELEKTRONIKI KOMPUTEROWEJ
Skr. pocztowa 35, 90-955 Łódź 8, tel. 57-25-83
ZEKOM
Oferuje użytkownikom systemów MERA-9150 i REDIFON
TERMINAL EKRANOWY
MR 1241
będący funkcjonalnym odpowiednikiem monitora MERA-7951.
Terminal MR 1241 przeznaczony jest do wprowadzania danych do systemu minikomputerowego MERA-9150.



JEŚLI INTERESUJE PAŃSTWA:

- profesjonalny sprzęt o wysokiej jakości, niezawodny w eksploatacji
- sprawny serwis
- krótkie terminy dostaw lub dostawy natychmiastowe

TO WYROBY ZEKOMU SĄ DO PANSTWA DYSPOZYCJI

dla użytkowników systemów MERA-9150 i REDIFON kompatybilny z MERA-7951

Nowe bloki systemu Multibus II

W roku 1985 pojawił się pierwszy wyprodukowany przez firmę Intel zestaw mikrokomputerowy zbudowany z elementów systemu Multibus II, tzw. **Evaluation Kit**, przeznaczony do wspomagania konstruowania sprzętu i przygotowywania oprogramowania. Składa się on z następujących modułów (por. Informatyka, nr 2–3, str. 44, 1986):
- boku centralnej obsługi CSM/001, przeznaczonego do wstępnego uruchomienia zestawu zgodnie z wymaganiami systemowymi,
- mikrokomputera iSBC 286/100, zawierającego mikroprocesor iAPX 286, z zegarem o częstotliwości 6 MHz,
- pamięci iSBC MEM/312, o pojemności 312 KB, łączonej z mikrokomputerem za pomocą magistrali LBX.

Blok centralnej obsługi zawiera – zgodnie z zaleceniami – układy zegarowe generujące sygnały taktowania magistrali, układy generowania sygnału alarmowego zaniku zasilania, sygnału zerowania, sygnału przekroczenia czasu transmisji, zegar astronomiczny i układy przypisywania numerów identyfikacyjnych blokom systemu, dołączonym do magistrali PSB. Ponadto zawiera również układy sprzęgające z systemem Multibus I.

Moduł mikrokomputera ma dostęp do magistrali równoległej PSB, magistrali lokalnej LBX i magistrali pomocniczej SBX oraz do wszystkich przestrzeni adresowych. Do mikroprocesora iAPX 286 dołączony jest koprocesor 80287, układ sterowania DMA, programowany układ wejścia-wyjścia o różnorodnych możliwościach pracy i dwa programowalne wejścia-wyjścia (np. RS232C i RS422A).

Moduł pamięci umożliwia operacje ze słowem 8-, 16- i 32-bitowym. W celu przyspieszenia dostępu do pamięci moduł ten zawiera 8 K-bajtową pamięć podręczną (ang. cache) typu RAM. Moduł pamięci jest dwuwejściowy, tzn. dostęp do zasobów pamięci jest możliwy przez magistralę LBX oraz magistralę PSB. Adresy początkowe i końcowe obszarów pamięci dostępnych z obu magistral są ustalone programowo.

Oprogramowanie zestawu składa się z monitora uruchomieniowego iSDM 286 (zapisanego w pamięci EPROM mikrokomputera) i pakietu wspamagającego iRMX 86II×SUPP.

Połączenie modułów mikrokomputera i pamięci z magistralą PSB jest zrealizowane za pomocą układów specjalizowanych, obsługujących protokół systemu – BAC 84110 (ang. bus arbiter controller) i MIC 84120 (ang. message interrupt controller).

Możliwości pierwszego zestawu mikrokomputerowego wykonanego w systemie Multibus II są bardzo ograniczone. Ponadto, jego początkowa cena była dość wysoka, a możliwości zakupu znacznie utrudnione ze względu na zastrzeżenia patentowe i handlowe.

Aktualne oferty Multibus II, np. firmy Siemens, są znaczenie bogatsze.

Pojawił się nowy typ bloku centralnej obsługi CSM-R500 zawierający jednocześnie sprzęg systemu ASM-M. Pojawiło się również wiele mikrokomputerowych bloków specjalnych, zawierających sprzęgi międzymagistralowe i międzysystemowe, np. iSBC LNK-001 zawierający:
- sprzęg Multibus II z Multibusem I i wewnętrzną pamięć pomocniczą o pojemności 128 KB, 16 M-bajtową pamięć Multibus I dostosowaną do wymagań Multibus II, 32 K-bajtową pamięć Multibus I dostosowaną do obszaru we-wy Multibus II i konwerter Multibus I na przerwania Multibus II;
- mikrokomputerowy blok FA-LANI ze sprzęgiem magistrali Bitbus zawierający mikroprocesory 8044 i Z80;
- mikrokomputerowy blok EUROLOG MB2--LNK-1 ze sprzęgiem systemu EUROLOG i jego magistrali wejścia-wyjścia (blok ma możliwość przeprowadzenia pełnej diagnostyki magistral, z którymi współpracuje, tzn. Multibus II, EUROLOG, we-wy).

Oferowane są nowe moduły mikrokomputerów, np. mikrokomputer iSBC 386/100 zawierający:
- mikroprocesor iAPX 386 pracujący z zegarem 16 MHz,
- koprocesor 80287,
- pamięć podręczna 64 KB,
- pamięć dynamiczną RAM z detekcją błędów parzystości o pojemnościach 1, 2, 4 lub 8 MB, z możliwością rozszerzenia do 16 MB,
- czterokanałowy układ 82586 DMA,
- sprzęg (8- lub 16-bitowy) magistrali SBX i DMA i magistralą wejścia-wyjścia,
- pojedyncze wyjście RS 232.

Inny mikrokomputer iSBC 186/100 zawiera:
- mikroprocesor iAPX 186 pracujący z zegarem 8 MHz i koprocesorem 8087 oraz mikroukładem MPC (ang message passing coprocesor), zapewniającym sprzężenie z magistralą zamiast układów MIC i BAC,
- pomięć pomocniczą o pojemności 512 KB, dwuwejściową,
- wewnętrzne układy testowania systemu,
- czterokanałowy układ DMA (82258),
- wyjścia RS232C i RS422A

Z kolei mikrokomputer MT 68020 jest oparty na układach firmy Motorola i zawiera:
- mikroprocesor MC 68020 pracujący z zegarem 12,5 lub 16,67 MHz, koprocesorem MPC i koprocesorem MC 68881,
- dynamiczną pamięć pomocniczą RAM o pojemności 1 MB,
- czterokanałowy układ DMA MC 68450,
- dwa wejścia-wyjścia szeregowe,
- sprzęg SBX,
- wyjście równoległe.

Oferowana jest również rodzina mikrokomputerów typu OSM B36, B26, B16 pracujących z mikroprocesorami iAPX 386, 286 lub 186, o rozbudowanych możliwościach programowych. Firma Siemens produkuje również mikrokomputer CP186-010 (z mikroprocesorem iAPX 186) przeznaczony do zestawów przemysłowych.

Bardzo liczna jest rodzina bloków pamięciowych systemu Multibus II, wykonanych przy użyciu różnych technologii. Przykładowo, oferowane są:
- pamięć domenowa typu M II-1 o dużej szybkości działania, mogąca pełnić rolę pamięci masowej,
- pamięć statyczna RAM-512 z magistralą LBX, wykonana z układów CMOS, pracująca ze słowem 32-bitowym,
- pamięć statyczna RAM 1000 do współpracy z magistralą PSB, wykonana w podobnej technologii,
- rodzina pamięci uniwersalnych SBC MEM (312, 310, 320, 340), z których każda ma pamięć podręczną o pojemności 8 KB (moduły pamięci są dołączane do magistral LBX i PSB, ich pojemności wynoszą 0,5, 1, 2, 4 MB),
- moduł pamięci uniwersalnej MEM-601 zawierający oprócz pamięci RAM możliwość wbudowania układów pamięciowych ROM, EPROM, NVRAM, EEPROM, dołączonej do magistral LBX i PSB,
- rodziny pamięci MSC 5910 o pojemnościach 1, 2, 4, 8, 16 MB, z pamięcią podręczną 32 KB,
- pamięć masowa MSC 6000 GIGARAM o pojemności 4 GB, mogąca pełnić rolę tzw. dysku półprzewodnikowego, pracująca ze słowem 32-bitowym, magistralami LBX i PSB, posiadająca bezpośrednie wyjścia RS 232 do testowania,
- rodzina pamięci OSM B11x o pojemności 0,5, 1, 2, 4 MB i OSM 12x o pojemności 1, 2, 4 MB, zawierających pamięci podręczne 8 KB i sprzęgi z magistralami PSB i LBX (pamięci te charakteryzują się dużą szybkością działania).

Pojawiły się również moduły mikrokomputerowe sprzężenia systemu z różnorodnymi urządzeniami zewnętrznymi, np.:
- sterowniki dysków elastycznych typu CD 22-4500,
- sterownik stacji taśm TAPE MASTER 2000 współpracujący z ośmioma jednostkami,
- sterownik urządzeń peryferyjnych SBC 186/224 sterujący cztery dyski stałe i cztery dyski elastyczne,
- szybki sterownik urządzeń peryferyjnych MB II ESDI 5201 sterujący czterema stacjami dysków,
- sterownik wielourządzeniowy OSM-B 340 do sterowania stacji dysków i urządzeń wejścia-wyjścia,

- 8-kanałowy sterownik komunikacyjny CD22-3800 (8×RS232 lub 6×RS232 i 2×RS422),
- sprzęg sieci Ethernet (IEEE 802.3),
- sterownik magistrali szeregowej OSM--B222,
- sterownik komunikacyjny ICC-M2 z mikroprocesorem MC 68020 i sprzęgami RS 232, RS 422, Ethernet.

Oferowany jest również szeroki asortyment modułów analogowo-cyfrowych, np.:
- wielofunkcyjny analogowo-cyfrowy blok wejścia-wyjścia typu RTI-915, posiadający 32 pojedyncze lub 16 różnicowych wejść analogowych z 12-bitowymi przetwornikami A/C i 4 wyjściowe kanały analogowe (częstotliwość przetwarzania przetwornika A/C – 71 kHz, a przetwornika C/A – 50 kHz; moduł jest sterowany cyfrowo),
- moduł wejścia-wyjścia typu 32 I/0 posiadający 32 wejścia i 32 wyjścia z optoizolacją,
- sterownik serwomechanizmów,
- sterownik urządzeń przemysłowych I/0 188-120 z mikroprocesorem iAPX 188 pracującym z zegarem 10 MHz, obsługującym 120 równoległych linii wejścia-wyjścia (podział 95 linii na wejście i wyjście jest ustalany przez użytkownika),
- sterownik mikroprocesorowych urządzeń przemysłowych o analogicznych możliwościach,
- sterownik wyjść analogowych A1 186--080 z mokroprocesorem iAPX 186 (10 MHz) sterującym 80 niezależnych kanałów wejściowych lub 40 różnicowych (i analogiczny moduł bez procesora),
- sterownik wyjść analogowych A0 186–032 z mikroprocesorem iAPX 186 (10 MHz) i 12--bitowym przetwornikiem cyfrowo-analogowym, posiadający 32 multipleksowane kanały wyjściowe (i analogiczny moduł bez procesora),
- 64-kanałowe moduły wejść cyfrowych,
- rodziny analogowych modułów wejścia--wyjścia MIB 186-140, MIB II 186/141, CESA 2/01, CESD 2/01,
- rodziny cyfrowych modułów wejścia-wyjścia MIB 186-111 i MIB II 186–130.

Ponadto produkowane są liczne moduły do współpracy z urządzeniami graficznymi.

W przedstawionym, krótkim przeglądzie najnowszych opracowań uwzględniono przede wszystkim propozycje firmy Siemens, która postawiła na bardzo dynymiczny rozwój systemu Multibus II. W ciągu roku od opracowania pojedynczych modułów systemu, powstało i jest sprzedawanych bardzo wiele typów modułów o bardzo rozległych możliwościach. Charakterystyczną ich cechą jest niezwykle wysoki stopień upakowania układów o wielkim stopniu scalenia w pojedynczym module. Większość modułów ma wewnętrzne systemy mikroprocesorowe kierujące pracą bloku, przy czym zwraca uwagę powszechne stosowanie mikroprocesorów 16- i 32-bitowych, pracujących z zegarami o dużej częstotliwości, oraz szerokie wykorzystanie koprocesorów, szczególnie koprocesorów MPC do obsługi protokółu magistrali. Zwraca uwagę duża liczba modułów pamięciowych i zarysowująca się tendencja eliminowania pamięci dyskowych.

Wydaje się, że rozwój systemu Multibus II jest uzależniony od możliwości technologicznych dostępnych producentom modułów. Ze względu na ogromny potencjał obliczeniowy nawet najprostszych zestawów można przypuszczać, że duże zestawy wielokomputerowe, do budowy których przewidziany jest Multibus II, jeszcze nie prędko będą tworzone. Ponadto wydaje się, że zapotrzebowanie na tak duże komputery jest ograniczone, natomiast budowanie zbyt prostych systemów w standardzie Multibus II jest nie ekonomiczne. Należy się spodziewać, że system Multibus II będzie pełnił rolę uniwersalnego nadrzędnego komputera nadzorującego pracę wielu mniej skomplikowanych systemów, co częściowo potwierdza się w przedstawionej ofercie firmy Siemens, zawierającej wiele bloków sprzęgających do systemów AMS-M, Bitbus i Multibus I.

KRZYSZTOF RZYMKOWSKI

Warunki prenumeraty na lata 1988–1989

Prenumeratorzy zbiorowi – jednostki gospodarki uspołecznionej, instytucje i organizacje społeczne zamawiają prenumeratę dokonując wpłat wyłącznie na blankiecie „wpłata–zamówienie" (jest to „polecenie przelewu" rozszerzone dla potrzeb Wydawnictwa o część dotyczącą zamówienia).

Blankiety te będą dostarczane dotychczasowym prenumeratom przez Zakład Kolportażu. Nowi prenumeratorzy otrzymają je po zgłoszeniu zapotrzebowania (pisemne lub telefoniczne) w Zakładzie Kolportażu.

Prenumeratorzy indywidualni – osoby fizyczne zamawiają prenumeratę dokonując wpłaty w UPT lub NBP na blankiecie NBP. Na odwrocie wszystkich odcinków blankietu należy wpisać tytuł czasopisma, okres prenumeraty, liczbę zamawianych egzemplarzy oraz wartość wpłaty. Wpłacać należy na konto: NBP III Oddział Warszawa 1036-7490--139-11.

Prenumerata ulgowa – przysługuje wyłącznie osobom fizycznym – członkom SNT, studentom i uczniom szkół zawodowych. Warunkiem prenumeraty ulgowej jest poświadczenie blankietu wpłaty (przed jej dokonaniem) na wszystkich odcinkach pieczęcią Koła SNT, wyższej uczelni lub szkoły. Sposób zamawiania prenumeraty ulgowej jest taki sam jak prenumeraty indywidualnej. W prenumeracie ulgowej można zamówić tylko po jednym egzemplarzu każdego czasopisma.

Uwaga! Miesięcznik „Aura" może być zamawiany w prenumeracie ulgowej również przez uczniów szkół ogólnokształcących.

Prenumerata ze zleceniem wysyłki za granicę – zamawia się tak jak prenumeratę indywidualną. Dodatkowo należy podać na blankiecie wpłaty nazwisko i dokładny adres odbiorcy.

Cena prenumeraty ze zleceniem wysyłki za granicę jest dwukrotnie wyższa.

Wpłaty na prenumeratę przyjmowane są w terminach:

- do 10 listopada na każdy kwartał, I i II półrocze oraz cały rok następny,
- do 28 lutego na II, III i IV kwartał oraz II półrocze,
- do 31 maja na III i IV kwartał oraz II półrocze,
- do 31 sierpnia na IV kwartał.

Zmiany w prenumeracie można zgłaszać pisemnie tylko w wyżej wymienionych terminach.

Informacji o prenumeracie udziela – Zakład Kolportażu Wydawnictwa NOT SIGMA (ul. Bartycka 20, 00–716 Warszawa), skr. poczt. 1004, 00–950 Warszawa, tel. 40--00-21 wew. 248, 249, 293, 297, 299 lub 40-30-86 i 40-35-89.

Egzemplarze archiwalne czasopism – można nabyć za gotówkę w Klubie Prasy Technicznej, Warszawa ul. Mazowiecka 12 (tel. 27-43-65), lub zamówić pisemnie. Zamówienia na egzemplarze archiwalne czasopism przyjmuje: Zakład Kolportażu, Dział Handlowy, 00–950 Warszawa, skr. poczt. 1004 (tel. 40-37-31), na rachunek dla instytucji lub za zaliczeniem pocztowym dla osób fizycznych.

Cena w 1988 roku

(dotyczy numerów 7–12)

CENA		Prenumerata normalna		Prenumerata ulgowa (bez zmiany)	
normalna	ulgowa (bez zmiany)	kwartalna	półroczna	kwartalna	półroczna
250 zł	50 zł	750 zł	1500 zł	150 zł	300 zł

Cena w 1989 roku

miesięczna		kwartalna		półroczna		roczna	
normalna	ulgowa	normalna	ulgowa	normalna	ulgowa	normalna	ulgowa
300 zł	60 zł	900 zł.	180 zł	1800 zł	360 zł	3600 zł	720 zł

Informatyka w Chinach

Pierwszy chiński komputer, Model 103 oparty na lampach elektronowych, ukończony był w 1958 roku w Pekińskim Instytucie Techniki Komputerowej z pomocą specjalistów radzieckich. Dwa lata później ZSRR wstrzymał pomoc techniczną dla Chin i przez następne dwie dekady Chińczycy musieli radzić sobie sami. Maszyny średniej generacji pojawiły się w 1965 r. Był to 32-bitowy komputer Model 109B i wolniejszy 21-bitowy komputer telekomunikacyjny – oba modele laboratoryjne, które nie weszły do produkcji.

W 1966 roku rozpoczęła się Rewolucja Kulturalna, co na dziesięć lat zahamowało rozwój nauki z informatyką włącznie. Jednakże pewne opracowania były prowadzone i na początku lat siedemdziesiątych zaczęły powstawać komputery oparte na obwodach scalonych, o różnej długości słowa, niezgodne z tendencjami światowymi. Nie widać też było wyraźnego upowszechnienia komputerów.

Punktem zwrotnym był rok 1978, gdy na Krajowej Konferencji Naukowej postanowiono, iż przyszła modernizacja Chin oparta będzie na nauce. Usprawnieniom techniki obliczeniowej nadano wysoki priorytet. Otwarte zostały kontakty z krajami rozwiniętymi i w ciągu ostatnich dziesięciu lat nastąpił znaczny rozwój informatyki chińskiej. W tym okresie sprowadzono z zagranicy ponad 1500 dużych maszyn i minikomputerów oraz ponad 150000 mikrokomputerów. Import w tej dziedzinie przekroczył w 1986 roku 300 milionów dolarów i był 4,2 razy większy niż w roku 1983, przy czym 63% tego importu pochodziło z USA, a 28% z Japonii.

Jednocześnie rozwinięto własny przemysł komputerowy. Niezależnie od fabryk produkujących mini i mikrokomputery podjęto prace nad automatyzacją rozpoznawania ideograficznych znaków języka chińskiego, a także doniesiono o opracowaniu procesora wykonującego 10 milionów rozkazów na sekundę i superkomputera „Rzeka Gwiazd" wykonującego 100 mln rozkazów na sekundę. Ze strony Chin jest to więc zmniejszanie opóźnień powstałych w stosunku do krajów rozwiniętych, ze strony firm zachodnich to penetracja ogromnego rynku obejmującego 1,1 mliarda ludzi.

Ocenia się, że obecnie w Chinach pracuje ponad 170 tysięcy zawodowych informatyków, z czego połowa w nowych zastosowaniach. Według danych z 1986 roku ponad 60% komputerów jest używanych w transporcie i przemyśle, 17% w nauce i nauczaniu, a tylko 3,4% w handlu i finansach, przy czym stwierdza się duże niewykorzystanie komputerów ze względu na brak wykwalifikowanych operatorów i odpowiedniego oprogramowania. Podaje się rząd 15–25% wykorzystania możliwości istniejących systemów.

Wśród firm prowadzących intensywny handel z Chinami należy wyróżnić IBM i Siemensa. Ta pierwsza stworzyła nawet specjalną filię w Hongkongu. Chiński odpowiednik PC nazywa się „Wielki Mur" i jest produkowany w 13 fabrykach w liczbie 25 tysięcy rocznie, wykorzystujący mikroprocesory Intela i kostki pamięciowe japońskiej firmy Oki. Produkowany jest komputer wielożytkowy, pracujący w systemie Unix BCMC 68000, wykorzystujący mikroprocesor Motoroli. Myśli się o produkcji odpowiednika DEC-owskiego MicroVAXa.

Istnieje około 2600 mieszanych spółek chińsko-zagranicznych, wiele firm zachodnich chciałoby w ten sposób doprowadzić do budowy swych zakładów montażu komputerów. Odbywają się też szkolenia dla użytkowników systemów instalowanych w Chinach, jak np. Honeywell Bull DPS 7000.

Jeśli chodzi o sieci komputerowe, to mimo podjęcia prac już w 1978 r. i wzrostu liczby pracujących sieci lokalnych, niewiele udało się zrealizować w zakresie sieci dalekosiężnych, ze względu na zły stan linii telekomunikacyjnych. Obecnie Chiny zgłosiły udział w akademickiej sieci Bitnet, do której miały być dołączone od 1 października 1987, a od czerwca 1988 sieć ta ma łączyć 17 węzłów lokalnych przez sieć krajową zwaną „Chinanet". Automatyzacja transmisji znaków języka ikograficznego stworzyła podstawy do zastosowań administracyjnych i biurowych. W 1986 roku ok. 30 tys. studentów chińskich studiowało za granicą, zapoznając się z zachodnimi metodami zarządzania i technikami analizy systemów. Największy wpływ na przyszły rozwój informatyki w Chinach będzie mieć prawdopodobnie współpraca z przedsiębiorstwami zagranicznymi.

Oprac.: J. RYŻKO
na podst. Datamation 17.1987

Kto jest kim w IFIP

Ping-Xin Guo

Prof. Ping-Xin Guo, od 1983 r. członek Zarządu IFIP, a od 1980 r. przedstawiciel Chin w Zgromadzeniu Ogólnym, urodził się w 1918 r. w Wu-Han. Po uzyskaniu na Uniwersytecie Centralnym niższego stopnia naukowego w dziedzinie nauk technicznych, pracował od 1954 r. jako inżynier elektronik. W 1954 r. został mianowany dyrektorem Departamentu Nauki i Techniki w Administracji Przemysłu Elektronicznego i był odpowiedzialny za realizację planu, w wyniku którego powstał pierwszy w Chinach komputer. W 1963 r. został mianowany wicedyrektorem Departamentu Nauki i Techniki w Ministerstwie Przemysłu Elektronicznego, odpowiedzialnym za komputery. W 1976 r. został wicedyrektorem Państwowej Administracji Przemysłu Komputerowego.

Obecnie prof. Ping-Xin Guo kieruje długofalowym planem przetwarzania informacji w języku chińskim, obejmującym systemy przetwarzania tekstów, szybkiego wyszukiwania informacji oraz składu i druku laserowego. Ostatni system, stosowany z powodzeniem w poligrafii, został wyróżniony nagrodą państwową pierwszego stopnia w dziedzinie nauki i techniki. Prof. Gou prowadzi też własne prace nad systemem rozpoznawania chińskich znaków.

Ping-Xin Guo zainicjował odpowiedni program i kierował projektowaniem i wytwarzaniem w Chinach systemów mikrokomputerowych. Założył również ogólnonarodowe przedsiębiorstwo usług informatycznych, w tym również programistycznych. W 1985 r. został mianowany przewodniczącym Komitetu Nauki i Techniki Komputerowej Ministerstwa Przemysłu Elektronicznego. Jest profesorem Uniwersytetu Beijing. Prof. Guo ma na swym koncie wiele publikacji, także książkowych, na temat nauki i techniki komputerowej.

Oprócz wykonywania swych podstawowych zajęć profesor jest kierownikiem rady Chińskiego Instytutu Elektroniki, wiceprzewodniczącym Chińskiego Towarzystwa Komputerowego i przewodniczącym Chińskiego Stowarzyszenia Informacji Medycznej.

Prof. Guo wraz ze swoją żoną Shi-Qionng, odpowiedzialną w Ministerstwie Przemysłu Elektronicznego za elektronikę medyczną, mieszka w Beijing razem z dwójką dzieci, które są również programistami. Do jego ulubionych rozrywek należy muzyka oraz tradycyjne chińskie sporty i ćwiczenia.

Oprac.: MK
na podstawie IFIP Newsletter

□ Wytwórca półprzewodników Signetics z Sunnyvale w Kalifornii rozpoczął masowe dostawy kostek EPROM wykonanych w technologii CMOS o pojemności 256 Kb. Kostki te, o nazwie 27C256 są zgodne z kostkami 27C64A tej firmy o pojemności 64 Kb, których dostawy trwają od lipca 1986 r. Mają one organizację 32 K × 8 bitów i kosztują 10 dolarów, przy zamówieniu 100 szt. Oba rodzaje kostek są wykonywane w ceramicznej obudowie 28-nóżkowej typu DIP z okienkiem kwarcowym.

□ Firma Maxtor Corp. z San Jose w Kalifornii opracowała nową stację dysków EXT-8760, o pojemności 760 MB. Pojemność tę osiągnięto dzięki równoczesnemu zagęszczeniu ścieżek i zapisu. Inne firmy, jak Priam Corp. i Micrpolis Corp., rozpoczęły dostarczanie pierwszych stacji dysków o pojemności 380 MB.

□ W Stanach Zjednoczonych rozpoczęła pracę pierwsza cyfrowa sieć usług zintegrowanych (ISDN – ang. *Integrated Services Digital Network*) w dyrekcji firmy MacDonalds, zajmującej się przemysłem spożywczym. Łączy ona około 300 stanowisk różnego rodzaju (operujących danymi, głosem i obrazem), wyposażonych w sprzęt firm AT&T, NEC i Fujitsu.

□ Przedsiębiorstwo VLSI Technology z San Jose w Kalifornii opublikowało właściwości swego systemu ekspertowego o nazwie Design Assistant. Umożliwia on użytkownikom optymalne rozmieszczanie, podział i dokonywanie zabiegów technicznych przy projektowaniu układów scalonych do specjalnych zastosowań (ang. *Application – Specific Integrated Circuit* – ASIC). Działa na ogólnych schematach blokowych dostarczonych przez użytkownika.

□ Firma National Semiconductor zapowiada wyprodukowanie mikroprocesora 32-bitowego o częstotliwości zegara 30 MHz. W kostce mikroprocesora będzie się znajdować także układ zarządzania pamięcią. Opracowywano też koprocesor, który uzupełni koprocesor zmiennoprzecinkowy produkowany przez firmę Weitek.

□ Według oceny firmy Motorola w 1986 r. rzeczywista stopa wzrostu światowego rynku elementów półprzewodnikowych wyniosła 5%. Na rok 1987 przewidywano, że wskaźnik ten osiągnie 17%. Dla poszczególnych rynków parametry te wyglądają odmiennie. W 1986 r. przemysł amerykański zanotował wzrost 7%, a w 1987 r. wzrost ten miał być równy wzrostowi światowemu i wynieść 17%. Japonia wykazała się w 1986 r. spadkiem sprzedaży o 3%, natomiast w 1987 r. przewidywano, że osiągnie najwyższy wskaźnik wzrostu wynoszący 19%. Dla Europy wreszcie, która w 1986 r. zanotowała aż 4% spadek, przewidywano na rok 1987 najniższy wzrost wynoszący 13%.

Wartość produkcji elementów półprzewodnikowych w 1986 r. wyniosła prawie 26,5 mld dolarów, z czego 10,6 mld dotyczyło Japonii, 8,6 mld – Stanów Zjednoczonych, 5,2 – Europy, a 2 mld reszty świata. Na rok 1987 przewidywano obroty światowe w wysokości 30,8 mld dolarów, z czego poszczególne rynki w wymienionej kolejności miały reprezentować odpowiednio: 34, 39, 19 i 8%.

W Europie, po dwóch kolejnych latach stagnacji, oczekuje się pewnego wzrostu. W 1986 r. RFN, Wielka Brytania i Francja reprezentowały 33%, 20% i 16% rynku europejskiego (w 1985 r. odpowiednio 30, 21 i 15%). Pierwsze dwa kraje wykazały wówczas spadek obrotu o 4%, a Francja o 5%. Przewidywano, że w 1987 r. 25-procentowy wzrost obrotów przemysłu samochodowego spowoduje zwiększenie zapotrzebowania na półprzewodniki. Najsilniej ma to wystąpić w przemyśle francuskim (14%), RFN (12%) i brytyjskim (10%).

W skrócie ● W skrócie ● W skrócie ● W skrócie ● W skrócie ● W skrócie

ELEKTRA

SPÓŁDZIELNIA PRACY

ul. Wąska 8, 81–659 Gdynia
Tel.: 241337, 241169
Teleks: 054520 spe pl.

Oferujemy WIELOWYJŚCIOWY, IMPULSOWY ZASILACZ DO URZĄDZEŃ KOMPUTEROWYCH, o następujących parametrach:

$U_1 = +5$ V, $I_1 = 1{,}5$ A
$U_2 = +8 \div 12$ V, $I_2 = 300$ mA
$U_3 = -6 \div 12$ V, $I_3 = 200$ mA

Zasilacz nasz posiada zabezpieczenie ponadnapięciowe i przeciwzwarciowe.
Informacji udzielamy pisemnie, telefonicznie lub teleksem.
Gwarantujemy konkurencyjne oceny, wysoką jakość i krótkie terminy realizacji zamówień.

EO/510/88

Język Ada

A. Burns, A. M. Lister, A. J. Wellings: A Review of Ada Tasking. Springer-Verlag, Berlin, 1987

Z powodu założeń projektowych Ada jest językiem złożonym. Posiada pascalopodobne konstrukcje typów i mechanizmy sterowania, mechanizm obsługi wyjątków, strukturę pakietu do przesłaniania informacji i dekompozycji programów, rozłączną kompilację, możliwość programowania na niskim poziomie, jak również zadaniowy model współbieżności. Ten ostatni obszar był chyba najczęściej dyskutowany i krytykowany.

Autorzy książki postawili sobie za cel dokonanie przeglądu modelu zadaniowego w świetle obszernych analiz i komentarzy, które pojawiły się w literaturze. Przegląd jest szeroki i obejmuje takie zagadnienia, jak:

- Ada jako uniwersalny, współbieżny język programowania,
- Ada jako język programowania systemów wbudowanych i rozproszonych,
- dokonane implementacje, ze szczególnym uwzględnieniem systemów rozproszonych,
- semantyka formalna, specyfikowanie i weryfikowanie,
- proponowane zmiany w języku.

Poruszenie tak wielu tematów w jednej publikacji jest bardzo pożyteczne dla potencjalnego użytkownika wielozadaniowości Ady, może on zapoznać się ze wszystkimi czynnikami mającymi wpływ na wydajność, niezawodność i poprawność tworzonego oprogramowania. Książka dostarcza wielu materiałów do dyskusji o modyfikacjach i tendencjach rozwojowych języka.

D. C. Luckham, F. W. von Henke, B. Krieg-Brückner, O. Owe: ANNA – A Language for Annotating Ada Programs. Springer-Verlag, Berlin, 1987

Anna jest rozszerzeniem Ady, pozwalającym dołączyć programy formalnego specyfikowania zamierzonego zachowania programów w Adzie. Zaprojektowanie Anny odpowiada potrzebie wzbogacenia Ady o dokładne, przetwarzane maszynowo anotacje. Dzięki temu, do programów w Adzie można stosować dobrze znane, formalne metody specyfikowania i dokumentowania.

Obecnie, projekt Anny obejmuje anotacje dla wszystkich mechanizmów Ady z wyjątkiem zadań. Podobne rozszerzenia formalnego specyfikowania mogą być stworzone dla innych algolopodobnych języków, takich jak Pascal, PL/I, Concurrent Pascal, Modula, a co istotne, te rozszerzenia byłyby podzbiorem Anny.

Projektując Annę, przestrzegano czterech zasad:

1. Tworzenie anotacji powinno być łatwe dla programisty i powinno zależeć w jak największym stopniu od notacji i koncepcji Ady.
2. Anna powinna posiadać cechy języka szeroko wykorzystywane przy specyfikowaniu i dokumentowaniu programów.
3. Anna powinna stanowić szkielet, w obrębie którego można stosować w odniesieniu do Ady różne uznane teorie formalnego specyfikowania programów.
4. Anotacje powinny być dobrze dostosowane do różnych zastosowań, podczas całego okresu istnienia programu. Takie zastosowania obejmują nie tylko testowanie, uruchamianie i formalne sprawdzenie kończonego programu, ale również specyfikowanie części programu, dokonane podczas wcześniejszych etapów analizy wymagań i projektowania.

G. L. Mohnkern, B. Mohnkern: Applied Ada. TAB Books, Chichester, 1987

W książce będącej wprowadzeniem do Ady położono szczególny nacisk na ogólne możliwości języka, bez wskazywania konkretnego kompilatora. Książka zawiera omówienie zarówno podstawowych właściwości języka, jak również właściwości zaawansowanych, takich jak operacje równoległe, rekurencja i in., ilustrowane przykładami użytecznych programów. Programy przykładowe, które czytelnik może dostosować do własnych potrzeb, ilustrują pewne pożyteczne aspekty programowania. Szczególną uwagę zwrócono na prezentację elementów Ady, przy użyciu formalizmów Backusa-Naura oraz procedury i funkcje CALCULAT, obejmujące także działania na liczbach zespolonych, operacje na macierzach i wektorach, zarządzanie bazami danych, wykorzystanie pragm do sterowania kompilacją i użycie pakietów rodzajowych.

A. N. Habermann, U. Montanari (Eds.). System Development and Ada. Springer-Verlag, Berlin, 1987

W książce przedstawiono szereg środowisk programowych dostępnych na rynku – opartych na metodach formalnych. Zbiór artykułów stanowi przegląd najważniejszych rodzajów środowisk dla Ady, stworzonych w ostatnich latach.

Spis treści: J. E. Archer Jr.: Projekt środowiska firmy Rational. F. Gallo: Inicjatywa PCTE – europejskie podejście do inżynierii oprogramowania. C. Z. Mitchell: Dostosowanie Ady na VAX-a wielojęzykowego środowiska programowego. V. Stenning: Środowisko Ady – pogląd osobisty. S. J. Westfold, L. Z. Markosian, W. A. Brever: Wytwarzanie oprogramowania z bazą wiedzy – od wymagań do kodu. E. Astesiano, G. Reggio: Podejście SMoLCS do semantyki formalnej języków programowania. D. Björner: Grafy projektów i meta-programy – w stronę teorii wytwarzania oprogramowania. C. B. Jones: Wytwarzanie oprogramowania oparte na metodach formalnych. B. Krieg-Brückner: Integrowanie procesów konstruowania i weryfikowania programów – metodologia PROSPECTRA. J. B. Goodenough: Atestacja kompilatorów Ady – przykład teorii i praktyki testowania oprogramowania. A. N. Habermann: Instytut Inżynierii Oprogramowania Uniwersytetu Carnegie-Mellon. D. C. Luckham, D. P. Helmbold, S. Meldal, D. L. Bryan, M. A. Haberle: Język tworzenia sekwencji zadań dla specyfikowania rozproszonych systemów Ady.

S. H. Saib, R. E. Fritz: Tutorial – The Ada Programming Language. Computer Society Press New York, 1987

Jest to zbiór wielu wczesnych artykułów na temat Ady i jej środowiska. Książka zawiera również artykuły o charakterze szkoleniowym, objaśniające najważniejsze aspekty języka, szczególnie znajdujące się we wbudowanych systemach czasu rzeczywistego – dziedzinie, dla której Ada jest przeznaczona.

M. KUC

Tylko prenumerata zapewnia otrzymanie każdego numeru

INFORMATYKI

System plików (2)

W drugiej części artykułu zajmę się terminami związanymi z implementacją systemu plików. Dla użytkownika i implementatora istotne są następujące pojęcia:

identyfikator pliku (file identifier) identyfikuje plik w systemie plików przez podanie alfanumerycznej nazwy pliku (file name) z rozszerzeniem (extension) określającym rodzaj pliku, niekiedy z numerem generacji (generation number) lub numerem wersji (version number) oraz identyfikatorem urządzenia (unit identifier), określającym urządzenie z nośnikiem pliku w strukturze hierarchicznej;

katalog[1] (directory) grupuje pliki w pojedynczym węźle struktury hierarchicznej; wyróżnia się katalog główny (root directory) oraz katalog użytkownika (user directory), a także katalog macierzysty (home directory) i podkatalog (subdirectory); identyfikator katalogu (directory identifier) jest nazwą katalogu będącą częścią opisu ścieżki dostępu;

utworzenie (create), **otwarcie** (open), **zamknięcie** (close, shut) i **usunięcie** (delete, erase) pliku są czynnościami systemu operacyjnego umożliwiającymi następnie dostęp do pliku;

wolumen lub wolumin (volume) jest zbiorem plików na danym urządzeniu (dysku stałym lub wymiennym, dyskietce, taśmie magnetycznej itp.) z określoną etykietą wolumenu (volume label) i logicznie montowany (mounted) w systemie operacyjnym;

formatowanie (format) wolumenu często nazywa się formatowaniem dysku lub dyskietki i polega na utworzeniu na nośniku jego logicznego podziału z zapisem fizycznym podstawowych struktur danych dla systemu plików;

prawa dostępu (access rights) oznaczają zakres możliwości (capabilites) wykorzystania danego pliku przez użytkownika lub grupę użytkowników do odczytu, zapisu, modyfikacji lub wykonania.

Omawianie organizacji wewnętrznej pliku zacznę od organizacji fizycznej (physical organization) pliku, której dotyczą następujące terminy:

blok fizyczny (physical block) jest ciągłym obszarem nośnika, możliwym do zaadresowania przy zapisie lub odczycie; blok fizyczny dla dysku odpowiada pojedynczemu sektorowi (sector); dla taśmy jest to fragment nośnika o stałej lub zmiennej długości;

blok logiczny (logical block) czasem nazywany klastrem (cluster) jest spójnym obszarem nośnika, równoważnym krotności jednostek alokacji (allocation unit), przyległych bloków fizycznych o długości nie większej od długości bufora w obszarze pamięci operacyjnej (przeznaczonego do przechowywania strumienia lub rekordów z danymi); dla taśmy magnetycznej blok logiczny odpowiada blokowi fizycznemu;

katalog deskryptorów (directory, catalog) jest specyficznym plikiem wykorzystywanym bezpośrednio przez system plików, a służącym do przechowywania deskryptorów (descriptor) plików użytkowych oraz podkatalogów (subdirectory); w deskryptorze jest umieszczona nazwa pliku, jego atrybuty, prawa dostępu, właściciel, czasy utworzenia i ostatnich modyfikacji oraz odwzorowanie relacji wiążących poszczególne elementy danych należące do danego pliku.

Kolejne terminy są związane z metodami rozmieszczania (allocation) bloków logicznych pliku na nośniku dyskowym. Wyróżnia się przydział spójny lub zwarty (contiguous) oraz niespójny (noncontiguous). Ten ostatni, chociaż łatwiejszy w realizacji, powoduje rozproszenie (distribution) danych pliku na nośniku. Przeciwną właściwością jest integralność (integrity) pliku, co po przydziałach niespójnych można uzyskać przez reorganizację nośnika (trwającą niekiedy nawet kilkanaście godzin!).

Do opisu zbioru bloków logicznych należących do danego pliku wykorzystuje się:

mapę bitową (bit map, block oriented file mapping), w której każdy bit odpowiada określonemu blokowi logicznemu, przechowując jego stan – wolny, zajęty;

tabelę bloków (block table, index block chaining), w której każda pozycja zawiera adres pewnego bloku fizycznego (w bloku logicznym) przydzielonego do danego pliku;

listy (lists, block chaining) wiążą w sekwencje bloki logiczne należące do danego pliku; ciekawą odmianą tej koncepcji jest tabela rozmieszczenia plików FAT (file allocation table) wykorzystywana w systemie MS-DOS.

Zależnie od przyjętej metody opisu zbioru bloków logicznych, na nośniku podczas formatowania określone bloki – przeważnie na ścieżce zerowej – są wypełniane opisem bloków logicznych uszkodzonych, wolnych lub zajętych, pozostałych na nośniku. Tworzony jest też katalog główny.

Na zakończenie zajmę się pojęciami występującymi przy implementowaniu systemu plików w ramach systemu operacyjnego. Moduł systemu plików jest warstwą (layer) systemu operacyjnego, podzieloną na następujące poziomy (level) obsługi:

organizacji fizycznej (physical organization) pliku realizujący polecenia transmisji określonej liczby bajtów z nośnika lub na nośnik, z uwzględnieniem organizacji fizycznej pliku;

organizacji logicznej (logical organization) pliku przekształcający polecenie dostępu do pliku na polecenie transmisji liczby bajtów, z uwzględnieniem organizacji logicznej pliku;

nazw symbolicznych (symbolic name), czyli zarządzania plikami (files management); na tym poziomie jest wykorzystywana tabela plików otwartych (open files table, file control block) przyspieszająca współpracę z plikami wcześniej otwartymi oraz kojarzącą nazwę z jego identyfikatorem logicznym.

W tym miejscu muszę stwierdzić, że nazewnictwo warstw oraz poziomów lub modułów ogólnie zwanych systemem plików (np. FILES-11) jest najczęściej lokalnym „wynalazkiem" tworzącej go firmy. Dlatego też podane wyżej trzy nazwy poziomów należy traktować raczej jako ogólny przykład. Może ktoś z Czytelników zaproponuje ściślejsze określenia.

WACŁAW ISZKOWSKI

[1] Nazywamy również skorowidzem (przyp. red.).

Ogłoszenia • Ogłoszenia

Programy, instrukcje i udoskonalenia techniczne dla komputerów ATARI, AMSTRAD, COMMODORE, IBM oferuje Agencja Komputerowa 41–200 Sosnowiec, P-157. tel. 63-29-35.
puterowa 41–200 Sosnowiec, P-157, tel. 63-29-35.

EO/423/87

Ogłoszenia • Ogłoszenia

Przypominamy, że 10 listopada upływa termin zamówienia prenumeraty INFORMATYKI na 1989 rok

NIE DAJEMY RECEPT
SPRZEDAJEMY NARZĘDZIA

ELEKTRONIKA – FILM – KOMPUTER

ul. Barska 2/20, 02–315 Warszawa

tel. 23-67-57, teleks 816955 efcom pl

OFERUJE:

* bogate profesjonalne oprogramowanie systemowe i narzędziowe umożliwiające tworzenie systemów użytkowych:
  - zarządzanie magazynami,
  - gospodarka materiałowa,
  - planowanie i kalkulacja,
  - lista płac,
  - finanse i księgowość,
  - symulowanie pracy przedsiębiorstwa,
  - przetwarzanie i edycja tekstów,
  - grafika dla celów zarządzania,
  - prowadzenie korespondencji

* wdrażanie systemów, szkolenie personelu oraz dostawy niezbędnego sprzętu mikrokomputerowego wraz z gwarancją i serwisem:
  - mikrokomputery IBM PC/XT/AT kompatybilne,
  - urządzenia peryferyjne,
  - wyposażenie

Sprzedajemy profesjonalne narzędzia
dla profesjonalistów

KONSULTACJE GRATIS

EO/592/88

Appelrath H-J.: System wyszukiwawczy dla płyty wizyjnej

INFORMATYKA 1988, Nr 7, s. 1

Charakterystyka systemu wyszukiwania informacji ODIR, zrealizowanego w Instytucie Informatyki Politechniki w Zurychu na mikrokomputerach osobistych IBM PC/XT, Lilith 1 oraz MacIntosh z wykorzystaniem laserowej płyty wizyjnej.

Appelrath H-J.: Information retrieval system for optical disc

INFORMATYKA 1988, No. 7, p 1

Characteristics of the ODIR information retrieval system, realized in the Informatics Institute of Zurich Technical University on IBM PC/XT, Lilith 1 and MacIntosh personal computers applying optical disc.

Appelrath H-J.: Ein Recherchesystem für Bildplatte

INFORMATYKA 1988, Nr. 7, S. 1

Eine Charakteristik des ODIR-Informationrecherchesystems, das im Informatik Institut der Technischen Universität Zürich auf IBM PC/XT, Lilith 1 und MacIntosh-Personalcomputern mit Anwendung einer Bildplatte realisiert wurde.

Błaszczak S.:Język Occam (1)

INFORMATYKA 1988, Nr 7, s. 4

Pierwsza część charakterystyki języka Occam przeznaczonego do programowania transputerów.

Błaszczak S.: Occam language (1)

INFORMATYKA 1988, No. 7, p 4

First part of characteristics of the Occam language, whisch is destined for transputer programming.

Błaszczak S.: Occam-Sprache (1)

INFORMATYKA 1988, Nr. 7, S. 4

Erster Teil einer Charakteristik von der Occam-Sprache für Transputer-Programmierung.

Fuglewicz J., Korniak P.: Projektowanie i implementacja wielodostępnego systemu operacyjnego na przykładzie systemu MASTER

INFORMATYKA 1988, Nr 7, s. 7

Założenia i sposób realizacji wielodostępnego systemu operacyjnego MASTER dla rozbudowanej konfiguracji minikomputera MERA-60, zrealizownego w Centrum Naukowo-Produkcyjnym Systemów Sterowania MERASTER w Katowicach.

Fuglewicz T., Korniak P.: Design and implementation of multiaccess operating system on example of the MASTER system

INFORMATYKA 1988, No. 7, p 7

Principles and realization method of the MASTER multiaccess operating system for extended configuration of MERA-60 minicomputer, which is realized in MERASTER Scientific and Manufacturing Center for Control Systems in Katowice.

Fuglewicz T., Korniak P.: Projektierung und Implementierung eines Mehrbenutzerbetriebssystems an Beispiel des MASTER-Systems

INFORMATYKA 1988, Nr. 7, S. 7

Voraussetzungen und Realisierungsmethode des MASTER-Betriebssystems, für erweiterte Konfiguration des MERA-60-Kleinrechners, das im MERASTER Wissenschaftlich-Produktionszentrum für Steuerungssysteme in Katowice realisiert wurde.

Zabrodzki J.: Cyfrowe generowanie obrazów (2)

INFORMATYKA 1988, Nr 7, s. 10

Druga część charakterystyki rozwiązań konstrukcyjnych cyfrowego generowania obrazów z wykorzystaniem techniki rastrowej.

Zabrodzki J.: Digital image generation (2)

INFORMATYKA 1988, No. 7, p 10

Second part of charakteristics of digital image generation hardware solutions applying raster technology.

Zabrodzki J.: Digitale Bildgenerierung (2)

INFORMATYKA 1988, Nr. 7, S. 10

Zweiter Teil einer Charakteristik von Hardwarelösungen der digitalen Bildgenerierung mit Anwendung der Rastertechnik.

Mokrzycki W.: Techniki obrazowania informacji w trójwymiarowych zdyskretyzowanych scenach

INFORMATYKA 1988, Nr 7, s. 13

Charakterystyka nowych metod i rozwiązań obrazowania informacji o obiektach trójwymiarowych.

Mokrzycki W.: Information display technics in diserete 3D scenes

INFORMATYKA 1988, No. 7, p 13

Characteristics of new methods and solutions for information displaying of 3D objects.

Mokrzycki W.: Methoden der Informationsdarstellung in der 3-dimensionalen diskreten Bühnen

INFORMATYKA 1988, Nr. 7, S. 13

Eine Charakteristik von neuen Methoden und Lösungen der Informationsdarstellung von 3-dimensionalen Objekten.

Abramowicz W.: Nowy sposób pamiętania w systemach informacyjnych

INFORMATYKA 1988, Nr 7, s. 17

Stan aktualny i perspektywy rozwoju pamięci optycznych.

Abramowicz W.: New way of storing in information systems

INFORMATYKA 1988, No. 7, p 17

To-day situation and development prospects of optical storages.

Abramowicz W.: Neue Speicherungsmethode für Informationssysteme

INFORMATYKA 1988, Nr. 7, S. 17

Eine Besprechung des heutigen Standes und der Entwicklungsaussichten im Bereich der optischen Speicher.

Bielecki J.: Turbo C. Włączanie warunkowe

INFORMATYKA 1988, Nr 7, s. 19

Metody i przykłady stosowania dyrektyw włączania warunkowego w języku Turbo C.

Bielecki J.: Turbo C. Conditional inclusion

INFORMATYKA 1988, No. 7, p 19

Methods and examples for conditional inclusion directives in the Turbo C language.

Bielecki J.: Turbo C. Bedingte Eingliederung

INFORMATYKA 1988, Nr. 7, S. 19

Methoden und Beispiele der Anwendung von Direktiven der bedingten Eingliederung in der Turbo C-Sprache.

Turowicz A.: Lokalna sieć komputerowa D-Link (2)

INFORMATYKA 1988, Nr 7, s. 21

Druga część charakterystyki sieci D-Link, zawierająca przykład praktycznego zastosowania tej sieci.

Turowicz A.: D-Link local area network (2)

INFORMATYKA 1988, No. 7, p 21

Second part of the D-Link network characteristics, which contains examples of practical application of the network.

Turowicz A.: D-Link-Lokalnetz (2)

INFORMATYKA 1988, Nr. 7, S. 21

Zweiter Teil einer Charakteristik von dem D-Link-Computernetz, der eine Besprechung eines Beispiels von praktischen Anwendung dieses Netzes umfasst.

6-letnie doświadczenie software'owe sprawdzone w ponad 1000 zakładach pracy

TYLKO **SYSTEM CSK/32** jeśli zarządzanie Twoim przedsiębiorstwem wymaga minikomputera o mocy obliczeniowej Odry oraz kilkunastu skomputeryzowanych stanowisk pracy

System CSK/32 to:

MIKROKOMPUTERY QUATRO CSK/32
— przystosowane do pracy z kilkunastoma terminalami, pracujące w dowolnym trybie graficznym

OPROGRAMOWANIE
— systemowe WDOS, MIKROLAN
— narzędziowe — MEGA BLOK, TABPLAN, PL-TEKST, BGRAF, TRYS
— aplikacyjne — FK K & K, EM K & K, KADRY, PŁACE

WDROŻENIA, SZKOLENIA
— zespoły ds. wdrożeń i szkoleń pracują w:
Gdyni tel. 248 018, tlx 054792 ● Krakowie tel. 373 573 ● Poznaniu tel. 676 271
Wrocławiu tel. 481 679 ● Warszawie tel. 258 528, tlx 816711
Sopocie — w wyspecjalizowanym Salonie Sprzedaży, ul. Kombatantów 1

CSK™

computer studio kajkowscy

EO/74/88

MIĘDZYWOJEWÓDZKA SPÓŁDZIELNIA PRACY „SIÓDEMKA"

ŁÓDŹ, AL. KOŚCIUSZKI 93,

ZAKŁAD INFORMATYKI I SYSTEMÓW KOMPUTEROWYCH

poleca po **najniższych cenach** w kraju:

— mikrokomputery 8-, 16-, 32-bitowe najwyższej jakości renomowanych firm z całego świata
— urządzenia peryferyjne:
- drukarki — również 24-igłowe i laserowe
- streamery
- napędy dyskowe 3", 5.25"
- monitory monochromatyczne, kolorowe, EGA, HEGA, VGA
- karty rozszerzenia pamięci
- kontrolery
- dyski twarde typu Winchester 20 MB, 40 MB, 60 MB, 80 MB

— systemy wielodostępne i lokalne sieci mikrokomputerowe:
- MULTI-LINK
- D-LINK
- XENIX

— materiały eksploatacyjne:
- dyskietki 5.25" MD2-D
- dyskietki 5.25" MD2-HD
- dyskietki 3" CF2
- dyskietki 3.5" MF 2DD
- taśmy barwiące do wszystkich typów drukarek STAR i NEC

Termin realizacji zamówień **natychmiast** po złożeniu zamówienia. **Bezpłatnie:** szkolenia, kursy, zestawy oprogramowania narzędziowego i użytkowego — przy dostarczeniu kompletnych systemów.

Proponujemy również na wszelkiego rodzaju mikrokomputery programy wspomagające zarządzanie przedsiębiorstwem:

— system finansowo-księgowo-kosztowy
— system zbytu i zaopatrzenia
— system technicznego przygotowania produkcji
— system materiałowy
— system kadrowy i kadrowo-płacowy (również dla pracowników akordowych)

Wymienione systemy pracują w wersjach sieciowych i wielodostępnych.

Wszelkich informacji udzielamy codziennie (oprócz sobót) w siedzibie Zakładu w Łodzi przy Al. Kościuszki 101, tel. 36-51-00, w godzinach 8—16.

EO/1005/87

SPÓŁKA z o. o.

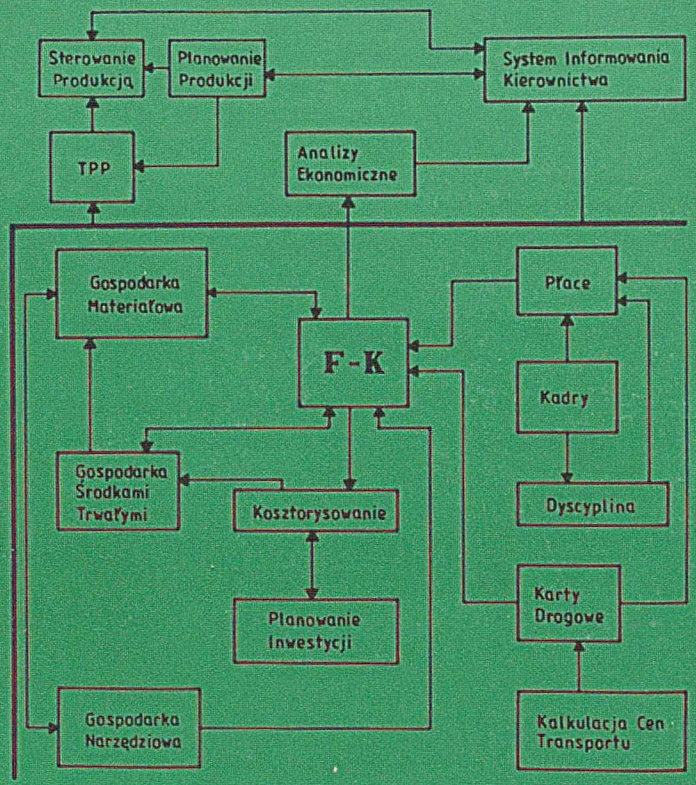
PRZEDSIĘBIORSTWO
INNOWACYJNO
WDROŻENIOWE

INTERTIMS

00–867 Warszawa, ul. Chłodna 35/37
tel.: 24 78 16, 24 78 23, teleks: 81 56 24

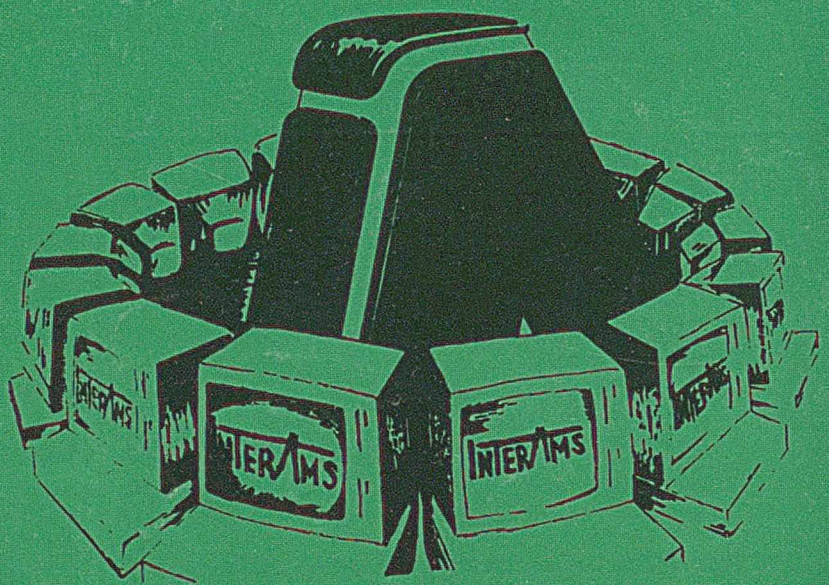
OFERUJE:

- *Zintegrowany pakiet systemów do zarządzania przedsiębiorstwem*
- *Usługi w zakresie kompleksowej komputeryzacji przedsiębiorstw*
- *Sprawdzone rozwiązania sieciowe np.: sieć „Novell"*
- *Kursy dla menadżerów*
- *Bank oprogramowania*
- *Sprzęt komputerowy najwyższej jakości z gwarancją do 1,5 roku*



ZAPRASZAMY...

do naszego salonu na prezentację sprzętu i oprogramowania



EO/390/88